RADC-TR-82-179
Final Technical Report
June 1982

AD A118479

SNEAK ANALYSIS APPLICATION GUIDELINES

Boeing Aerospace Company

Davey L. Buratti and Sylvia G. Godoy

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

82 08 23 069

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-82-179 has been reviewed and is approved for publication.

APPROVED: Bruce W. Dudley

BRUCE W. DUDLEY
Project Engineer

APPROVED: David C. Luke

DAVID C. LUKE, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBER) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-82-179
2. GOVT ACCESSION NO.: AD-A118479
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): SNEAK ANALYSIS APPLICATION GUIDELINES
5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 3 Feb 81 - 12 Mar 82
6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Davey L. Buratti, Sylvia G. Godoy
8. CONTRACT OR GRANT NUMBER(s): F30602-81-C-0053
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Boeing Aerospace Company, 1300 Bay Area Boulevard, Houston TX 77058
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 23380247
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (RBER), Griffiss AFB NY 13441
12. REPORT DATE: June 1982
13. NUMBER OF PAGES: 234
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): Same
18. SUPPLEMENTARY NOTES: RADC Project Engineer: Bruce W. Dudley (RBER)
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

Sneak Analysis	Criticality
Sneak Circuit	Effectiveness
Software Sneak Analysis	Design Concern
Application Guidelines	Drawing Error

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Sneak analysis is an engineering tool which can be used to identify designed-in conditions that can be inhibit desired system functions resulting in lower operational reliability. The objective of this study was to develop guidelines for applying sneak analysis to hardware and software contracts in a cost effective manner. Specific guidelines for establishment of sneak analysis requirements, monitoring of sneak analysis efforts, determining program costs, evaluation criteria and determining candidate

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

equipments have been defined. Criteria for tailoring the guidelines to the program needs have also been established. A measure of sneak analysis effectiveness was determined which shows excellent results in early program development phases including improved system reliability.

ABSTRACT

Sneak Analysis is aimed at identifying designed-in conditions that could inhibit desired system functions or produce undesired system functions which could adversely affect crew safety or mission equipment. The Sneak Analysis Application Guidelines effort established the technique as a cost-effective analysis tool which can be scheduled in reliability program plans specified in MIL-STD-785B. The analysis technique differs from other systems analysis techniques in that it is based on identifying designed in inadvertent modes of operation and is not based on failed equipment or software. Sneak Analysis can be effectively blended with fault related analyses to identify design and fault related problems in a cost-effective manner. The analysis produces excellent results in early program development phases, and can be used effectively to identify late development phase problems missed by testing and other analyses. System reliability is improved by the resolution of the Sneak Analysis identified problems.

CONTENTS

CONTENTS

CONTENTS

Appendices

FIGURES

TABLES

TABLES

ACRONYMS

ADCN	Advanced Design Change Notice
ASPR	Armed Services Procurement Regulation
BLS	Bureau of Labor Statistics
C3	Command, Control, Communication
CCFS	Common Cause Failure Analysis
CDR	Critical Design Review
CEI	Contract End Item
CON	Conceptual Phase
CPFF	Cost-Plus-Fixed-Fee
CRIT	Criticality
DCN	Design Change Notice
DCR	Design Concern Report
DER	Drawing Error Report
DID	Data Item Description
DoD	Department of Defense
DSARC	Defense Systems Acquisition Review Council
EC	Evaluation Criteria
EMF	Electro-Magnetic Force
EMI	Electro-Magnetic Interference
FACI	First Article Configuration Inspection
FFP	Firm-Fixed-Price
FHA	Fault Hazard Analysis
FMEA	Failure Mode and Effects Analysis
FSED	Full-Scale Engineering Development
FSPD	Full-Scale Prototype Development
FTA	Fault Tree Analysis
G&A	General and Administrative
HA	Hazard Analysis
HOL	High Order Language
IC	Integrated Circuit
LSI	Large-Scale Integrated Circuit
MSI	Medium-Scale Integrated Circuit
PDR	Preliminary Design Review
PHA	Preliminary Hazard Analysis
PP	Pilot Production Phase
PRDR	Preproduction Reliability Design Review
QC	Quality Control
RFP	Request for Proposal
ROM	Rough Order of Magnitude
SA	Sneak Analysis
SCA	Sneak Circuit Analysis
SDCR	Software Design Concern Report
SDER	Software Drawing Error Report
SHA	System Hazard Analysis
SOW	Statement of Work
SSA	Software Sneak Analysis
SSHA	Subsystem Hazard Analysis
SSI	Small-Scale Integrated Circuit
SSR	Software Sneak Report
UNP	Unlimited Production Phase
VAL	Validation Phase

SECTION 1

1. INTRODUCTION

Sneak Analysis is an engineering analysis tool which can be used for hardware and software systems to identify latent paths which cause concurrent unwanted functions or inhibit desired functions, assuming all components and codes are functioning properly. The objective of this analysis tool is the prediction of these problems before they occur in test or operation. If these problems are identified early in a program, the cost of modifications and redesign should be reduced and the reliability and safety of the system should increase. MIL-STD-785B (Reliability Program for System and Equipments Development and Production, 15 September 1980) has a task (#205) which, if specified, will require Sneak Analysis on new efforts. The number of dollars that can be allocated for reliability program tasks is limited for equipments and systems that are procured with fixed or constrained budgets. Therefore, it was necessary to determine whether Sneak Analysis was both effective and had estimable costs before inclusion as a program requirement. Sneak Analysis data have been collected and analyzed to provide proper visibility of costs and effectiveness. This effort resulted in the collection of these data, analysis in detail of the cost and problem identification effectiveness of Sneak Analysis, and development of guidelines for application of Sneak Analysis requirements.

1.1 Objective. The objective of this effort was to develop guidelines for applying and structuring Sneak Analysis based on the type of equipment or software, complexity, development phase and program dollars. These guidelines define Sneak Analysis program requirements, monitoring criteria and cost effectiveness parameters. These guidelines can be used to tailor requirements for Sneak Analysis to the needs of each individual program.

1.2 Approach. The approach for this effort was based on an in-depth study of 102 completed Sneak Analysis projects (out of a total of 111 projects) for various systems, equipment, software codes, and project environments. The study compared methods of analysis, cost of analysis, size and complexity of equipments, type of equipment, phase of development and effectiveness of the analysis. Based on these results, guidelines were developed for specifying, applying, and monitoring Sneak Analysis requirements. In addition, a feasibility study was performed for developing simplified Sneak Analysis techniques for small-scale hardware applications.

1.3 Problems. The acquisition of data associated with real program system change costs was very limited and difficult to obtain. There was little, if any, composite data at the program level which identified engineering, equipment, test, documentation and retrofit costs for approved changes. The statistical validity of program savings associated with correcting Sneak Analysis identified system problems is based on derived "average" project change costs. These change costs can vary significantly from project to project and have a resultant effect on postulated program savings.

1.4 Factor Index Guide. The major document topics are outlined in Table 1-1 according to primary and secondary references. The table should be of value to the reader in locating particular subject matter because of the document structure and size. The document structure is organized by the six tasks described in Section 2. The structure is based on the step-by-step Statement of Work tasks and the resulting engineering analysis effort. The size of the document is due in part to the extensive effort devoted to identifying Sneak Analysis effectiveness based on the actual project results, as presented in Appendix A.

TABLE 1-1. FACTOR INDEX GUIDE

TOPICS	PRIMARY FACTOR REFERENCE SECTIONS	SECONDARY FACTOR REFERENCE SECTIONS
APPLICATION GUIDELINES	3.5.6	
MONITORING GUIDELINES	3.4.4	3.4.2.1, 3.4.2.2
RISK ASSESSMENTS	3.5.1, 3.5.2, 3.5.4	
CONTRACTING PRACTICES	3.4.2.3	3.4.4.1, APPENDICES C-H
SPECIFICATION REQUIREMENTS	3.4.1	APPENDIX I
TAILORING PROCESS	3.4.3.1	3.5.5, 3.4.2.1
REQUEST FOR PROPOSAL CONSIDERATIONS	3.4.2.1	
EVALUATION CRITERIA CONSIDERATIONS	3.4.2.2	
QUALITY ASSURANCE	3.4.1.7	3.5.1, 3.4.4.4
HARDWARE ANALYSES	3.3.1, 3.3.2	3.4.3.1
SOFTWARE ANALYSES	3.3.3, 3.3.4	
PROBLEM IDENTIFICATION EFFECTIVENESS	3.2.1.2, 3.2.2.7, 3.2.3, 3.2.4.4	2.2, 3.4.4.5, 3.6.2.1, 3.6.2.2
COST EFFECTIVENESS	3.2.2.7, 3.2.3, 3.5.3	2.2.2, 3.2.2.1, 3.2.2.2, 3.2.2.3, 3.2.2.4, 3.2.2.5, 3.2.2.6
COST COMPARISONS	3.2.2.1, 3.2.2.2, 3.2.2.3, 3.2.2.4, 3.2.2.5, 3.2.2.6, 3.2.2.7, 3.2.3	
COST ESTIMATING	APPENDIX B, 3.5.6	3.4.3.1, 3.5.6, 3.6.1
PROJECT PHASING	3.2.1.1, 3.2.1.2 3.2.1.3, 3.2.1.4, 3.2.2.2	3.5.2, 3.5.3, 3.5.4
EQUIPMENT SELECTION	3.2.1.3, 3.2.1.4, 3.2.2.4, 3.2.2.6, 3.2.4 (ALL)	3.5.5
SYSTEM CRITICALITY	3.2.1.4, 3.2.2.5, 3.2.4 (ALL)	
PROJECT DURATION	3.2.2.3	2.2.3
FEASIBILITY STUDY	3.6	
PROJECT HISTORY TABLES	APPENDIX A	

2. SNEAK ANALYSIS APPLICATION SUMMARY

This section contains a summary presentation of the task flow, analysis trends based on actual project results, analysis technique comparisons, Sneak Analysis Application Guidelines, tailoring requirements, cost estimation techniques, and overall project conclusions. The detailed presentation of task material is presented in Section 3.

2.1 Task Flow. The Sneak Analysis Application Guidelines task flow is shown in Figure 2-1.

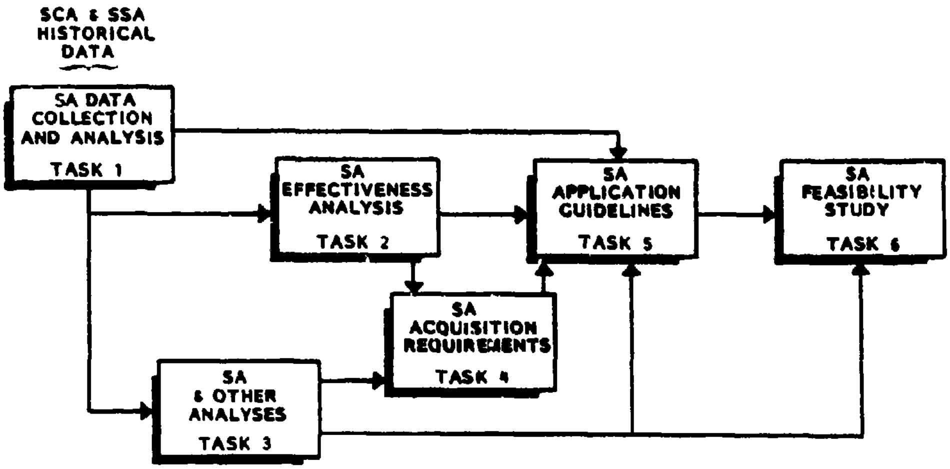


Figure 2-1. Task Flow

Task 1 required the collection of historical data for a statistically significant number of hardware and software Sneak Analysis projects. The information gathered represented the complete data base for one contractor acting in the role of an independent analyst. The 111 projects were sufficient for this project application and were the basis for the subsequent application guidelines effort. The importance of Task 1 is depicted in Figure 2-1 as it provides direct input to three tasks. The detailed historical data are presented in Appendix A. Task 1 results are presented in Section 3.1.

Task 2 required a detailed analysis of the historical data for relevant trends in terms of cost, technique effectiveness, project phasing, and types of equipment/software analyzed. This was the major task element of the total effort since the tables were to be used as the basis for any assertions or claims concerning Sneak Analysis. Task 2 results are presented in Section 3.2.

Task 3 resulted in an examination of various analysis techniques for commonalities and differences in an attempt to determine the most applicable tools for project use. Twelve hardware techniques and five software techniques were included in the study. Task 3 results are presented in Section 3.3.

Task 4 provided the Sneak Analysis requirements, specifications, sample Statements of Work, Request for Proposal requirements, evaluation criteria, and decision making processes. An additional task effort addressed the description of guidelines and roles of the procuring activity in Sneak Analysis projects. Task 4 results are presented in Section 3.4.

Task 5 resulted in the development of guidelines for the application of the Sneak Analysis tool. The task addressed cost, schedule, equipment, risk, and analysis effectiveness. This task, along with Tasks 3 and 4, provides the primary user output to be considered in the Sneak Analysis selection process. Task 5 results are presented in Section 3.5.

Task 6 produced the feasibility study results which addressed the development of Sneak Analysis capability for small-scale system applications. The feasibility study resulted in the conclusion that an automated Sneak Analysis program system could be developed which provides limited Sneak Analysis capability for small-scale systems. General system requirements are provided for this computerized system. Task 6 results are presented in Section 3.6.

2.2 Effectiveness Measures. Sneak Analysis has been used effectively in virtually all types of electrical and electronic systems and recently in software systems. Table 2-1 presents an abbreviated list of the Appendix A project systems and subsystems where hardware and software Sneak Analyses have been implemented.

TABLE 2-1. SNEAK ANALYSIS EQUIPMENT/SOFTWARE APPLICATIONS

SNEAK ANALYSIS EQUIPMENT/SOFTWARE APPLICATIONS	
ELECTRICAL POWER GENERATION	CAUTION/WARNING SYSTEM
ELECTRICAL POWER DISTRIBUTION	MONITOR/CONTROL SYSTEM
ELECTRICAL POWER CONTROL	INSTRUMENTATION SYSTEM
ELECTRICAL SUPPORT EQUIPMENT	DATA ACQUISITION AND PROCESSING
FLIGHT/LAUNCH SEQUENCER	TELEMETRY/SIGNAL CONDITIONING
FLIGHT CONTROL SYSTEM	DATA RECORDER
GUIDANCE/NAVIGATION SYSTEM	COUNTERMEASURE/DISPENSER SYSTEM
LANDING SYSTEM	LASER SEEKER SYSTEM
ATTITUDE CONTROL SYSTEM	ENVIRONMENTAL CONTROL SYSTEM
ANGLE-OF-ATTACK TRANSMITTER	EJECTION SEAT SEQUENCER
PROPULSION SYSTEM	DIAGNOSTIC SYSTEM
ENGINE CONTROL SYSTEM	LIGHTING SYSTEM
THRUST REVERSER/ FULL CONTROL	SAFETY SYSTEM
AVIONICS SYSTEM	TRANSPORTATION SYSTEM
THERMAL CONTROL	EXPERIMENT SYSTEM
ORDNANCE/PYROTECHNIC SYSTEM	COMPUTER DATA LINK CONTROLLER
ARMING AND FUSING SYSTEM	SHOP TEST EQUIPMENT
WEAPON CONTROL SYSTEM	BLOWOUT PROTECTION SYSTEM
FIRE CONTROL RADAR SYSTEM	TOWER LOWERING SYSTEM
DETECTOR SYSTEM	ON-BOARD SOFTWARE

The project applications have included the Space, Airborne, and Ground/Water environments. The applications have spanned from a total electrical system analysis to analysis of selected functions within a single system. Projects have been performed for mature technology systems to the more recent state-of-the-art systems. The applications have included complex system designs, extensively redesigned systems, systems with unresolved test problems and systems with field related problems. Table 2-1 represents a minimum listing of systems which can be considered candidates for future analysis, either separately or in combination.

In general, the larger the system(s), the more numerous are the equipment/software interfaces and the greater the number of problems identified in the areas included in the Sneak Analysis project. Size of system to be analyzed and cost of analysis are related since the cost for Sneak Analysis is based on the number of components for hardware systems and executable instructions for software systems. Analysis results are also highly influenced by program development phase and criticality of the system(s) analyzed.

2.2.1 Program phase. In order to determine the most effective development phase to implement Sneak Analysis, the relative program costs required to correct system problems by phase were developed and presented in Section 3.5.2. This information is essential to adequately develop the reliability program plan referenced in MIL-STD-785B. Sneak Analysis cost by phase and, more importantly, the postulated cost to modify equipment/software design can be predicted. This enables an orderly reliability program plan development, analysis tool planning and selection, and resource estimation and allocation. Figure 2-2 presents the change cost trendlines for hardware and software systems. Note the especially high relative program cost for changes in the late development phases.

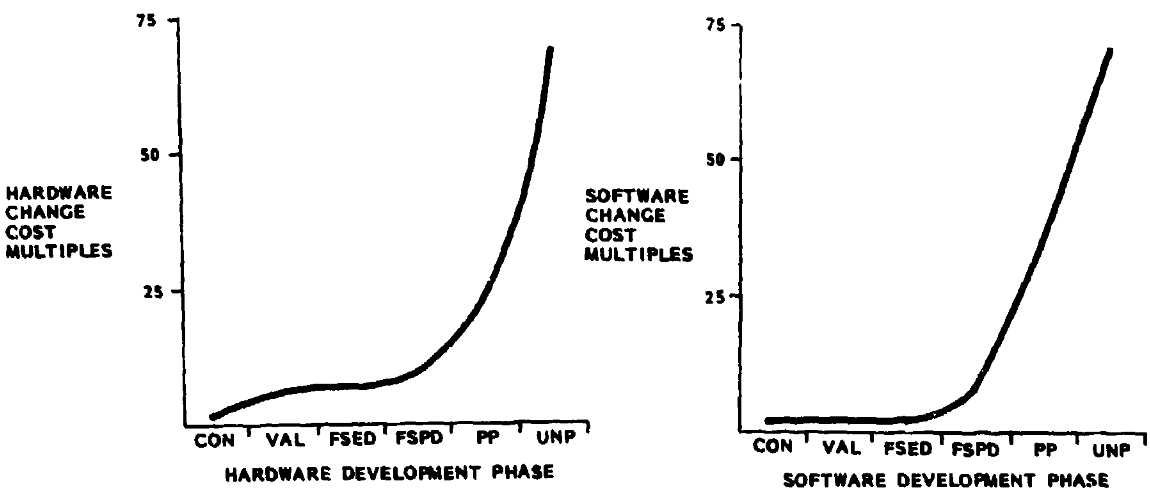


Figure 2-2. Development Phase Change Cost

The earlier the program development phase that a problem is identified and corrected, the lower the overall program cost. In Figure 2-2, the ideal analysis start times are the Conceptual (CON), Validation (VAL), Full-Scale Engineering Development (FSED), and Full-Scale Prototype Development (FSPD) phases. Since Sneak Analysis is based on detailed system drawings and computer program instructions, the most likely early phases for implementation would be the Full-Scale Engineering Development and Full-Scale Prototype Development phases. The costs associated with correcting problems in the latter two phases, Pilot Production (PP) and Unlimited Production (UNP), are significantly higher than the preceding four development phases. Change costs in the latter two phases can range from 10 to 100 times those of the earlier development phases. As such, the implementation of an analysis tool such as Sneak Analysis should be planned in the early development phases when the design is still reasonably fluid and can be changed in a cost-effective manner. Cost savings for early identification of problems appears to be potentially greater for software systems than for hardware systems.

One of the primary report findings resulting from the detailed study of Sneak Analysis effectiveness is that significant levels of equipment/software problems are present in systems, regardless of the program development phase. If these problems are not identified and corrected, operational reliability will be adversely affected. The definitive trend of analysis technique effectiveness in terms of the number of problem reports released is presented in Figures 2-3 and 2-4.

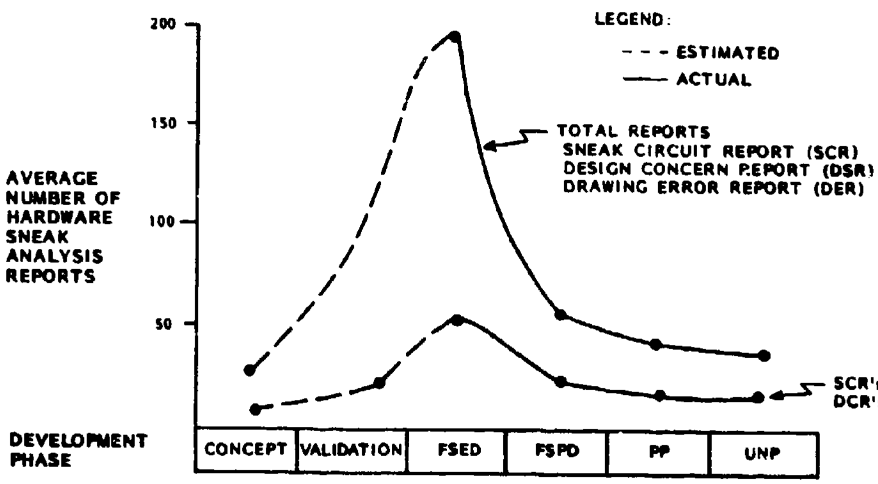


Figure 2-3. Development Phase Distribution of Hardware Reports

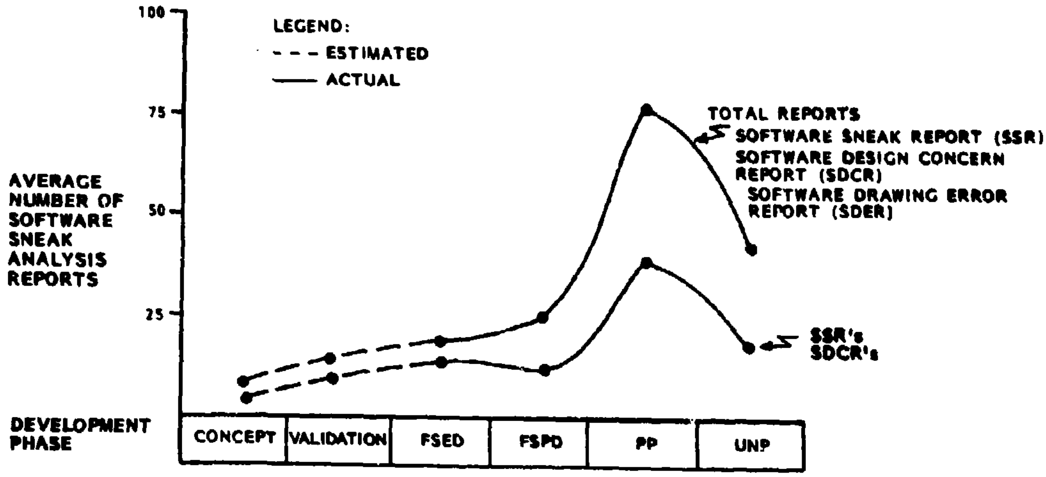


Figure 2-4. Development Phase Distribution of Software Reports

The hardware project report averages presented in Figure 2-3 illustrate the relative effectiveness of the Sneak Analysis technique in identifying system problems. The level of problem identification is very pronounced during the early development phases when it is most cost-effective to implement design changes. Although the report level declines in later development phases, it is very important to note that even mature and mass produced or one-of-a-kind systems still contain a sufficient level of embedded problems to justify performing the analysis. Projects in the Unlimited Production phase especially have been analyzed by other techniques, and thoroughly tested, and yet problems are still identified by the Sneak Analysis technique.

Figure 2-4 illustrates the corresponding effectiveness of Sneak Analysis in identifying software problems. The trendline shows a pronounced report level in the Pilot Production phase. The results are heavily influenced by application of Sneak Analysis on complete and delivered software, as opposed to analysis implementation during the earlier design phases. Based on actual project data, the greatest number of software problems are identified when the entire software code is integrated for test and operational usage (equivalent to the latter stages of verification and validation). As Software Sneak Analysis is implemented on future projects and if it becomes an accepted tool in verification and validation efforts, the effectiveness trendline should shift from the later development phases to the earlier development phases and thereby resemble the Figure 2-3 trendline. The identification and correction of only one serious hardware or software problem can offset the cost of Sneak Analysis. If the problem is found in an earlier development phase such as FSED or FSPD there can be an actual program cost savings.

2.2.2 Analysis cost. The volume and type of Sneak Analysis reports have a close correlation to contract value. Figure 2-5 illustrates the increasing report levels associated with the Sneak Analysis project values for hardware and software systems.

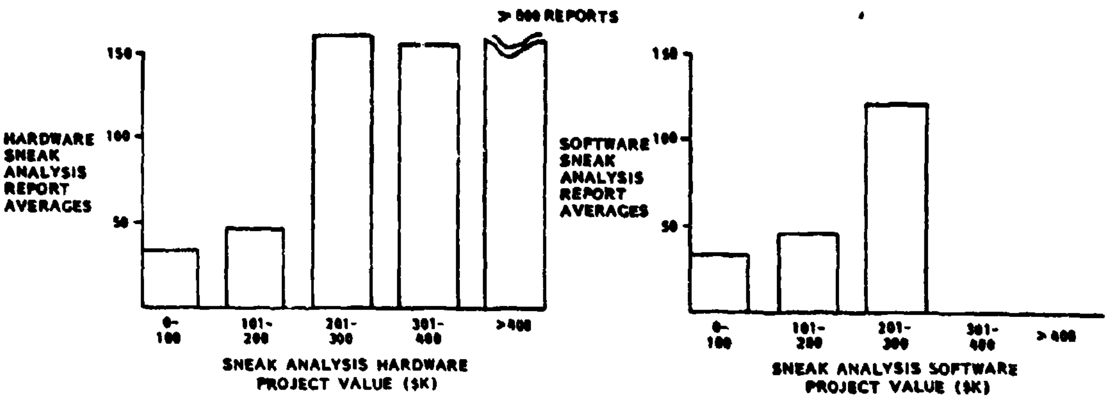


Figure 2-5. Project Value Report Levels

The cost of the analysis for hardware systems is based on the type (complexity) and number of electrical components, while software systems are based on the type (complexity) of computer program language and number of executable statements. The cost per component and program instruction is presented in Appendix B. The higher Sneak Analysis project values involved a larger number of systems as well as larger systems, that is, a larger number of components and computer instructions. Report levels and types of reports appear to be greatest for the higher contract value projects when large or complete systems are included in the task scope. The higher the contract value, the greater the number of systems (components and instructions) included in the analysis, and consequently the greater the report level. Report levels are significant, however, in the lower contract value range, which accounts for 75% of the historical Sneak Analysis projects.

Average cost for a Sneak Analysis project presented in Appendix A was 0.06% of the overall program cost. The highest percentage of Sneak Analysis cost to program cost was 0.4%, and the lowest was 0.0001%. While the level of problem reporting is dependent on analysis contract value, overall cost at the program level appears to have little if any predictable effect on the number of reports generated in a project analysis effort. More problems are typically embedded in large systems, but the scoping of a majority of the Sneak Analysis tasks to portions of these large systems obscures any definable program level trends.

2.2.3 Period of performance. The period of performance for a Sneak Analysis project to some extent is determined by the project value. Based on past performances, the trendline for Sneak Analysis project duration is as shown in Figure 2-6, which is based on the average project duration in each cost range.

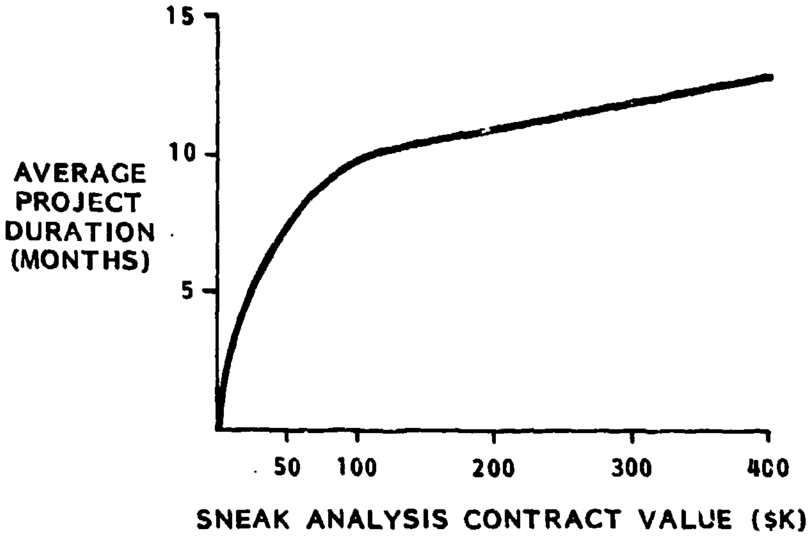


Figure 2-6. Sneak Analysis Period of Performance

The project duration for small applications is proportionately greater than large applications because of the required task steps involved, which are discussed in Section 3.4.2.2. Data acquisition and network tree preparation are the two main required task elements which influence small project duration, followed by system partitioning, encoding and data processing. For large projects, the trendline follows an apparent linear relationship.

2.2.4 Equipment classifications. The predominant types of equipment included in hardware Sneak Analysis projects are relay logic and digital systems, as shown in Figure 2-7. The relay classification includes resistors, capacitors, single load devices, diodes and switches. The digital classification includes discrete digital devices and complex integrated circuit devices. The analog classification includes amplifiers, inverters, converters, and feedback systems.

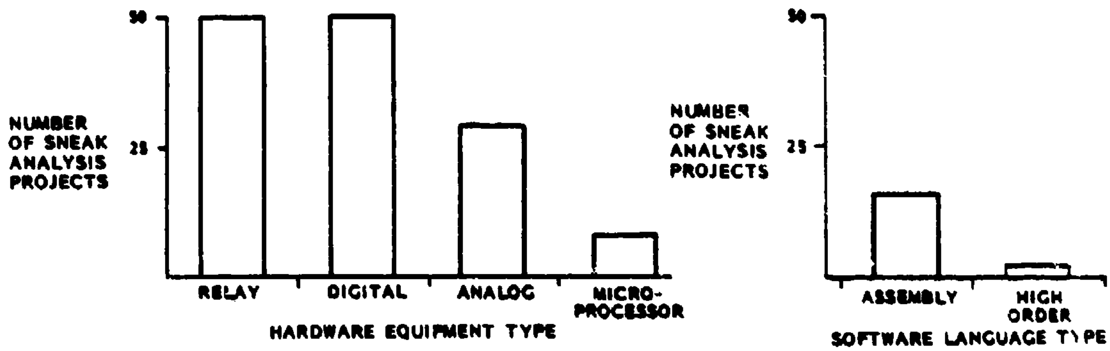


Figure 2-7. Sneak Analysis Equipment/Software Project Distribution

The majority of early Sneak Analysis projects were relay logic hardware systems. Current trends indicate a decided shift of analysis projects to systems involving digital, analog and microprocessor based systems for hardware. The early digital Sneak Analysis projects were for systems involving discrete devices, while the current trend is toward systems composed of one or more integrated circuits. Software projects are predominantly assembly language applications, but high order language projects are increasing. While Figure 2-7 might indicate the division of Sneak Analysis projects into single equipment/software type categories, the typical project is composed of a blend of equipment types.

In one-third of the hardware Sneak Analysis projects, the contracted effort involved a blending of two or more analysis techniques to focus on the orderly identification of system problems inherent in the design and also produced by equipment faults or failures. These analyses were conducted in a complementary manner. Sneak Analysis produced the system circuit diagrams and identified non-failure related conditions designed into the system. The associated fault

analyses were then based on an evaluation of critical paths and functions identified by the Sneak Analysis task. Very favorable results were produced in the combined analysis projects, including lower project costs.

The relative effectiveness of the Sneak Analysis technique with regard to the equipment/software criticality also displays a very significant trend, as shown in Figure 2-8. Projects involving man-in-the-loop (Criticality I) systems have much higher report averages than projects rated as mission critical (Criticality II) or systems that can cause mission degradation (Criticality III). These projects typically include a larger number of systems, including a greater number of equipment interfaces. The trend is very pronounced in hardware systems and moderate in software systems. Although the report averages are higher for Criticality I systems, significant report levels are present in Criticality II and III systems. It should be noted that the analyst performs the same analysis steps at the same depth regardless of the system criticality.

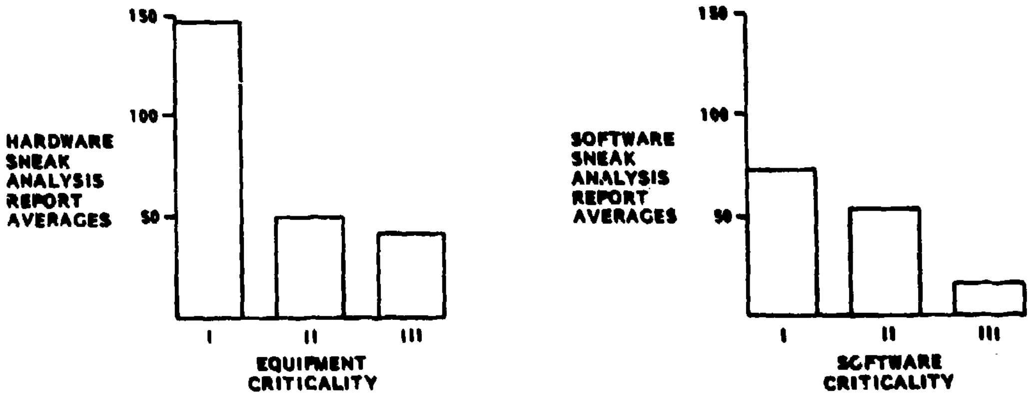


Figure 2-8. System Criticality Report Levels

2.3 Analysis Technique Comparison. Sneak Analysis is a unique analysis technique used to systematically identify component and software problems. Sneak Analysis has certain similarities to other analysis techniques and distinct differences to these same analysis techniques. A brief summary of the similarities and differences between Sneak Circuit Analysis and other hardware analysis techniques is presented in Table 2-2, while an equivalent comparison is presented in Table 2-3 between Software Sneak Analysis and software analysis techniques. Section 3.3 presents descriptions for each of the techniques contained in the two tables.

2.4 Application Guidelines. This section summarizes the Sneak Analysis Application Guidelines. The primary guidelines include:

1. Establishing need for Sneak Analysis

   a. Reliability improvements in the overall program result from the identification and resolution of system problems.

   b. Independent analysis is currently the only established approach for the analysis.

   c. Problem detection to eliminate the need for costly retrofits or redesigns and possible loss of irreplaceable systems such as spacecrafts or particular airborne equipment are immediate considerations for performing the analysis.

   d. High criticality systems (crew and mission critical) also warrant the analysis. Low criticality systems may be eliminated from consideration as long as no active control functions are performed in these systems.

   e. Unresolved system problems that have not been found by other analyses or tests are also good candidates for Sneak Analysis.

   f. A high change rate in the baseline design can also be used to justify the analysis.

   g. Sneak Analysis is a cost-effective tool in all phases of program development, but the analysis results exhibit a pronounced effectiveness in early development phases, and particularly in the Full-Scale Engineering Development Phase.

2. Determining applicable systems

   a. Systems which perform active functions are the primary candidates for Sneak Analysis. Electrical power, distribution and controls have traditionally been the main areas for hardware analysis. Computer programs which actively control and sequence system functions are good software candidates.

TABLE 2-2. HARDWARE ANALYSIS COMPARISON MATRIX

COMPARISON ELEMENT \ ANALYSIS TECHNIQUE	SNEAK CIRCUIT ANALYSIS	FAILURE MODES AND EFFECTS ANALYSIS	FAULT TREE ANALYSIS	COMMON CAUSE FAILURE ANALYSIS	SENSITIVITY ANALYSIS	WORST CASE ANALYSIS	[illegible] AND LOAD ANALYSIS	GROUNDING ANALYSIS	ACCIDENT ANALYSIS	PRELIMINARY HAZARD ANALYSIS	OPERATION HAZARD ANALYSIS	FAULT HAZARD ANALYSIS
PHASE												
CONCEPTUAL										X		
VALIDATION			X							X	X	
FSED	X	X	X		X	X				X	X	X
FSPD	X	X	X	X	X	X			X	X	X	X
PILOT PRODUCTION	X	X	X	X	X	X	X	X	X		X	X
UNLIMITED PRODUCTION	X	X	X	X	X	X	X	X	X		X	X
MODE												
MANUAL	X	X	X							X	X	
SEMIAUTOMATIC	X	X	X	X		X	X	X	X			X
AUTOMATIC					X							
EQUIPMENT												
RELAY	X	X	X	X			X	X	X			X
ANALOG	X	X	X	X	X	X						X
DIGITAL	X	X	X	X		X						X
ANALYSIS TYPE												
QUANTITATIVE			X	X	X	X	X	X	X			
QUALITATIVE	X	X	X	X	X	X	X	X	X	X	X	X
USE OF RESULTS												
PLANNING										X	X	
OPTIMIZATION					X							
OPERABILITY	X	X			X	X					X	
RELIABILITY	X	X			X	X						X
GRACEFUL DEGRADATION		X	X	X			X	X	X			X
PERFORMANCE UNDER STRESS		X	X	X			X	X	X	X		X
FAULT ISOLATION	X	X	X	X		X		X				
DESIGN EFFICIENCY	X				X							
DATA TYPE												
BLOCK DIAGRAM			X							X	X	
ENGINEERING DRAWINGS			X		X						X	X
INTEGRATED SCHEMATICS	X	X	X		X	X				X	X	X
PRODUCTION DRAWINGS	X	X	X	X	X	X	X	X	X		X	X
WIRE LISTS	X	X	X	X			X	X	X			
MECHANICAL DRAWINGS				X			X		X			X
CONNECTOR PIN MAPS				X			X	X	X			
PHOTOGRAPHS								X	X			
PROBLEM TYPES												
SNEAK PATH	X		X				X	X	X			
SNEAK LABEL	X										X	
SNEAK INDICATOR	X										X	
SNEAK TIMING	X				X	X						
OVER STRESSED COMPONENT	X				X	X	X		X			
WRONG COMPONENT	X				X	X	X		X			
COMPONENT FAILURE	X	X	X	X					X			X
LINE SHORT		X		X			X	X	X			
OUT OF BOUNDS TEMP				X					X	X	X	
MOISTURE LEAKAGE				X			X	X				

TABLE 2-3. SOFTWARE ANALYSIS COMPARISON MATRIX

COMPARISON ELEMENT / ANALYSIS TECHNIQUE	SNEAK SOFTWARE ANALYSIS	DESK CHECKING	PEER CODE REVIEW	STRUCTURAL ANALYZER	PROOF OF CORRECTNESS
Type	• Program Source Listing • Design Specifications • Logic Flow Diagrams • Language Description	• Program Source • Design Specifications	• Program Source Listing • Design Specifications	• Program Source Listing • Job Control Statements	• Program Source Listing • Design Specifications
Mode	Semi-automatic	Manual	Manual	Semi-automatic	Manual
Language Type	Assembly/HOL	Assembly/HOL	Assembly/HOL	HOL (2)	Assembly/HOL
Verification & Validation Phase Application	• Requirements/Specifications Analysis • Design Analysis • Code Evaluation • Verification Testing • Validation Testing	• Code Evaluation	• Code Evaluation	• Code Evaluation • Verification Testing	• Code Evaluation
Additional Cost to Software to Perform Task	4 - 40%	25%	10 to 50%	25 to 500%	100 to 500%
	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies • Identifies Discrepancies between Program Code and Program Documentation	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies • Verifies Output Values per Specifications	• Identifies Code Inefficiencies	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies	• Identifies Discrepancies between Program Code and Specifications

b. Passive systems that do not affect the overall program operation can be omitted from analysis consideration.

c. Sneak Analysis can and has been successfully implemented on complete vehicle or program applications, as well as limited subsystem or functional applications. The analysis is best performed on configurations involving numerous system interfaces and large size systems.

d. The applicable systems should be completely specified by component or instruction level documentation in the form of schematics, drawings, wire lists and source computer program code so that the analysis can be conducted at the "as-built" and "as-coded" levels, respectively.

e. Detailed analysis of critical systems can be performed by blending various analysis techniques which bring to bear the best features of each analysis in identifying design and fault related problems.

3. Scheduling requirements

a. Sneak Analysis should be scheduled so that final project results are obtained and can be adequately evaluated by the procuring activity and equipment manufacturers prior to the end of the Full-Scale Prototype Development Phase.

b. The preferred start time is prior to CDR in the Full-Scale Engineering Development phase. Optional change analysis should be considered to track the resulting system changes brought about by CDR.

c. Timely results can be obtained for all scheduled Sneak Analysis projects and also for those projects which are intended to identify a single test, operational or fleet problem. For single problem oriented Sneak Analyses, limited system scoping and available documentation can provide project results as soon as one to two months into the project schedule.

d. Orderly scheduling of Sneak Analysis can be based on the average four to six month project duration.

2.5 Tailoring Requirements. This section contains brief descriptions of the Sneak Analysis tailoring process, RFP and SOW considerations, and guidelines for monitoring task activities. The items are:

1. Tailoring process - When cost and schedule considerations effectively constrain a broad system application of Sneak Analysis, the procuring activity may reduce the scope of the effort to a smaller number of systems, components and functions. The tailoring process will reduce the quantity of hardware components and software instructions, but this should not affect the depth of the analysis. Particular care

must be exercised to ensure that the desired functions identified as in-scope are complete and adequately documented. Sneak Analysis can also be combined with other analyses to achieve an even greater reduction in overall cost for required analyses.

2. Proposal considerations

   a. Sneak Analysis Specification requirements (hardware and software) are presented in Section 3.4.1 and Appendix I. The specifications are developed for performing Sneak Analysis at the detailed component and program instruction level.

   b. Request for proposal considerations referenced in Section 3.4.2.1 have been developed, which identify and describe the various tasks involved in the Sneak Analysis process. These items are intended to provide the procuring activity with necessary and sufficient project requirements, competitive and sole-source considerations, and the fundamental analysis approach. Table 2-4 presents the outline for the RFP and evaluation criteria.

TABLE 2-4. OUTLINE OF A SNEAK ANALYSIS REQUEST FOR PROPOSAL AND EVALUATION CRITERIA

REQUEST FOR PROPOSAL (RFP) OUTLINE	
1. PROGRAM NAME 2. PURPOSE OF RFP 3. SCOPE OF EFFORT 4. APPLICABLE SUBSYSTEMS 5. ANALYSIS DEPTH 6. CHANGE ANALYSIS OPTION 7. ADDITIONAL ANALYSES	8. DATA REQUIREMENTS 9. TASK DESCRIPTIONS 10. DELIVERABLES 11. MISCELLANEOUS 12. FACILITIES & SECURITY 13. COST 14. TIME REQUIREMENTS
EVALUATION CRITERIA (EC)	
1. UNDERSTANDING PROBLEM 2. RELEVANT PAST PERFORMANCE 3. CAPABILITY TO PERFORM	4. COST 5. SCHEDULE

c. Evaluation criteria are provided in Section 3.4.2.2, which will aid the procuring activity in evaluating contractor responses to the RFP's and selecting the contractor to perform the analysis. Important criteria are applicable contractor experience, intended approach, depth of analysis, and cost.

d. A majority (84%) of the Sneak Analysis projects have been awarded as sole source Firm Fixed Price contracts. Cost-Plus-Fixed-Fee contracts are awarded for long duration Sneak Analysis projects, large system analyses, and those projects with optional change analysis.

3. Statement of Work requests - Example SOW's and associated instruments have been developed and are presented in Appendices C through H. The information includes:

   a. Hardware Sneak Analysis
   b. Software Sneak Analysis
   c. Integrated Hardware/Software Sneak Analysis
   d. Data Item Description
   e. Third Party (Proprietary) Data Working Agreement
   f. Project Schedule
   g. Combining Sneak Analysis with Other Analyses

4. Procuring activity monitoring guidelines

   a. Data acquisition has customarily been assigned to the procuring activity, otherwise extra cost is incurred. Proprietary data from vendors and contractors typically requires data agreements which may require significant time to acquire.

   b. Sneak Analysis report evaluation and coordination at problem review boards and engineering change boards are an important procuring activity function.

   c. Liaison, contract monitoring, contract modification and project closeout are the remaining procuring activity functions.

2.6 Cost Estimation Techniques. This section provides a brief method to estimate cost for Sneak Analysis projects and also provides several indicators which relate project cost to program level cost. The calculation of project cost and allocation of program budget considerations are as follows:

a. The cost of Sneak Analysis is computed on the basis of the number and type of hardware components and the number and type of computer program language instructions. Table 2-5 is used in cost computations and assumes the performance of a detailed hardware Sneak Analysis for all of the components in the estimate. Table 2-6 provides an example calculation of project cost for a typical system. Software Sneak Analysis cost is approximately $10 per assembly language instruction.

TABLE 2-5. COST FACTORS FOR DIFFERENT PART TYPES

PART TYPE	WEIGHTING FACTOR	WEIGHTING FACTOR TOLERANCE	PERCENT TOLERANCE
	$/PART	$/PART	
RESISTORS, CAPACITORS, COILS	29	±8	≈28%
RELAYS, TRANSISTORS, SWITCHES	79	±11	≈14%
SMALL-SCALE INTEGRATED CIRCUITS (SSI)	164	±14	≈ 9%
MEDIUM-SCALE CIRCUITS (MSI)	284	±14	≈ 4%
LARGE-SCALE INTEGRATED CIRCUITS (LSI)	468	±25	≈ 5%
GENERALIZED COMPONENT MIX (USED WHEN ACTUAL COMPONENT MIX IS NOT KNOWN)	94	±19	≈20%

TABLE 2-6. SAMPLE CALCULATIONS

	NUMBER OF PARTS	X	WEIGHTING FACTOR	=	COMPONENT COST		COST TOLERANCE
RESISTORS, CAPACITORS, COILS	400	X	29/PART	=	$ 11,600	≈	±3200
RELAYS, TRANSISTORS, SWITCHES	200	X	79/PART	=	15,800	≈	±2200
SSI	150	X	164/PART	=	24,600	≈	±2200
MSI	100	X	284/PART	=	28,400	≈	±1400
LSI	50	X	468/PART	=	23,400	≈	±1200
TOTALS	1,000				$103,800	≈	±10,200

b. Sneak Analysis can be scoped to individual systems, subsystems, and functions. Excessive scoping, however, could limit the analysis effectiveness by eliminating the detailed function tracking which is typically developed across system boundaries.

c. Sneak Analysis can be contracted for on an incremental basis on one or more of the higher criticality systems for which documentation is readily available, and concluding with the additional systems as data for these systems become available.

d. The procuring activity can expect annual program costs for Sneak Analysis <u>and</u> problem resolution to range from 0.1% in the early development phases to approximately 5% in the later phases. There are significant cost and risk penalties associated with late identification and resolution of system problems.

e. The ratio of Sneak Analysis cost to total change cost ranges from approximately 50% in the Concept phase to 0.5% in the Unlimited Production phase.

f. The percentage of Sneak Analysis cost for the entire program duration averages approximately 0.06% for the Space, Airborne and Ground/Water environments, with the highest level at 0.4% and the lowest level at 0.0001%.

g. Space Environment correction costs are the highest overall for the three environments, while the Ground/Water Environment has the highest single phase correction cost during Unlimited Production.

h. Program budget for the analysis should be allocated in the formulation of the reliability program plan and maintained throughout the development cycle for the desired schedule start time.

i. Since Sneak Analysis can be effectively blended with other analyses, reduced project costs for the combined analyses can be achieved.

2.7 <u>Conclusions</u>. Guidelines for application of Sneak Analysis have been developed and presented throughout this document with the aim of informing prospective and current project procuring activities about the nature, function, and roles of Sneak Analysis, which is Task 205 in MIL-STD-785B. The guidelines present pre-contract considerations, contracting methods, analysis scheduling, cost estimation, system applicability, expected results, and task monitoring activities. A thorough reading of the document will provide the procuring activity with the knowledge to effectively contract and manage a Sneak Analysis effort.

SECTION 3

3. TASK DETAILS

This section contains the detailed task requirements and results. The task requirements are reproduced from the SCA Application Guidelines Statement of Work. The task results presented are the detailed text materials, charts, figures, and tables generated during the course of this effort. The material included is intended to show the basis of the trends, requirements and descriptions for Sneak Analysis Applications.

The first five tasks represent the Application Guidelines effort while the sixth and final task represents a feasibility study effort.

3.1 Task 1 - SCA Data Collection and Analysis. Task 1 - Collect and analyze data on previously performed Sneak Circuit Analyses for hardware and, if possible, software efforts. The data to be collected shall include: the system/equipment nomenclature, the program contract dollar value, the phase of development, the sneak circuit requirements, the Sneak Circuit Analysis costs, the Sneak Circuit Analysis results and effectiveness, the types and complexity of equipments or systems to which the analysis was applied, and the criticality of the mission of the equipment or system. The data shall be collected for a statistically significant number of equipments for ground, airborne, and space environments. This does not preclude the use of engineering design judgment relative to equipment types not included in the available Sneak Circuit Analysis data base. These equipments shall be selected to be representative of current design technology.

This task required the collection of pertinent data on the 111 Sneak Analysis projects performed by The Boeing Company from 1967 through March 1981. The information is included in tabular form in Appendix A of this report. The projects have been chronologically divided within four main program environments including:

1. Space
2. Airborne
3. Ground/Water
4. Exclusions

The report formats for the Project History Tables contain the following category descriptions:

1. Project. The vehicle or project name for the Sneak Analysis task. When designated portions of the vehicle or project were analyzed, the subsystems were identified in the Equipment/Subsystem Requirements category.

2. Program Contract Value. This is the overall contract dollar level for the program or vehicle, not the designated subsystems. Program contract values were extracted from various sources, including:

a. 1980 DMS Market Intelligence Reports
b. U. S. Military Aircraft Data Book, 1978
c. U. S. Missile Data Book, 1980

The contract value is a rough indicator of program cost which includes research, development, test and procurement. The contract values presented are applicable as of the date of award of the Sneak Analysis contract.

In some instances the cost shown is for an entire line of equipment because the cost for the single configuration being analyzed could not be determined from the total. Where project costs could not be determined, an estimation of cost is provided and is flagged by an asterisk (*).

3. Equipment/Subsystem Requirements. This category contains the systems or subsystems, equipment interfaces, software, experiments, test equipment and other miscellaneous boundaries that are considered to be within scope of the Sneak Analysis task. Only the primary areas of the analysis are identified.

4. Equipment Criticality. Numerical values which indicate the criticality of the system/subsystem analysis. The values are:

   a. I - Loss of Life
   b. II - Loss of Mission
   c. III - Mission Degradation

5. Equipment Classification. The C^3 designation which provides a high level indication of system function:

   a. Command
   b. Control
   c. Communication

6. Equipment Type. This is a broad categorization of the equipment or software under analysis. For hardware systems, the entries were:

   a. Relay logic. Circuitry composed of relays (inductive loads with accompanying switch contacts). This category of equipment also includes display panels with manually operated switches and operator display devices.

   b. Digital. This involves two-state discrete and integrated circuit components.

   c. Analog. Circuitry which processes continuous functions for varying voltages and currents.

   d. Microprocessor. Typically a digital system which controls the operation and timing of a system based on input software.

   For software systems the entries were:

   a. High-Order Language. A programming language whose statements are translated into more than one machine language instruction. Examples are FORTRAN, COBOL and PLI.

b. Assembly Language. A symbolic form of machine language with instruction mnemonics and operands. In general, one statement in assembly language corresponds to and is translated into one machine language instruction.

7. Development Phase. This is the phase of the project development or production at the time of the Sneak Analysis procurement. The development phases may be referenced in Figure 3-1, which depicts the DoD Acquisition Phasing. Two abbreviations used are FSED (Full-Scale Engineering Development) and FSPD (Full-Scale Prototype Development).

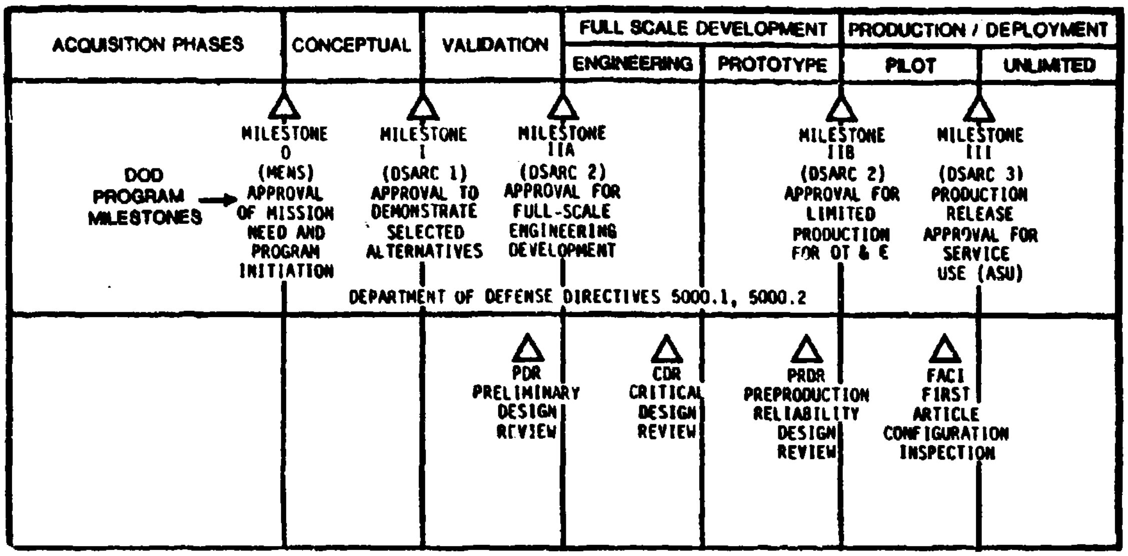


Figure 3-1. DoD Acquisition Phasing

8. Type of Analysis. This is an identification of the analyses performed. Many of the analyses shown are single functions, such as hardware or software Sneak Analysis. Some Sneak Analysis projects, however, involve a blend of analyses types, including Failure Modes and Effects Analysis (FMEA), Fault Tree Analysis (FTA), Common Cause Failure Analysis (CCFA), Fault Hazard Analysis (FHA), and Preliminary Hazard Analysis (PHA), also referred to as a Gross Hazard Analysis (GHA). Interface Analysis between hardware and software systems is also included.

9. Sneak Analysis Contract Dollars. This is the total contract value for the performance of the analysis effort. If the type of analysis category indicates multiple hardware analysis techniques, the listed cost is for all of the analyses combined.

10. Reports. Included in this category are the numbers and types of reports issued during the Sneak Analysis effort. The reports are the primary outputs of the analysis effort. The acronyms used for this category are:

a. SCR - Sneak Circuit Report
b. DCR - Design Concern Report
c. DER - Drawing Error Report
d. SSR - Sneak Software Report

e. SDCR - Software Design Concern Report
f. SDER - Software Drawing Error Report

The first three acronyms are for hardware related equipment or documentation, while the last three acronyms are related to software or software documentation.

The SCR's and SSR's are conditions identified in the system (hardware/software) which will inhibit the occurrence of a desired function or will generate the occurrence of an undesired function, without regard to equipment or software failure.

The DCR's and SDCR's are conditions identified in the system which could affect performance and reliability, or where undesirable design practices have been found.

The DER's and SDER's are conditions identified within the documentation (e.g., schematics, wire lists, procedures, listings) supplied by the building contractors or agencies.

11. Dates and Period of Performance. The year(s) and number of calendar months denoting the period of performance are included in this category. The contract initiation year is important to the Program Contract Value category and to the Development Phase category.

3.2 <u>Task 2 - Detailed Study of SCA Effectiveness</u>. Task 2 - Perform a detailed study using the data collected to determine the overall effectiveness of a Sneak Circuit Analysis considering cost of performing the analysis, the type, complexity, mission criticality, phase of development, and environment of the equipments or systems.

Analyses shall be performed to equate effectiveness (number of sneak paths, timing errors, drawing errors, etc. discovered) of the analyses to depth, complexity and costs of the analysis required. All assumptions used in this analysis shall be defined and justified.

Information in the Appendix A Project History Tables has been organized chronologically by the application environments of the hardware and/or software projects. The hardware/software composition of the projects, including the program environments, is displayed in Figure 3-2.

The total sample of this analysis effort consisted of 111 Sneak Analysis projects. Nine of these projects are listed in the Project History Table under the heading of Exclusions. These projects are either classified, proprietary or contain written agreements which limit distribution of results. There are a total of 102 Sneak Analysis projects which are distributed and reported in the remaining Project Tables. The distribution and categorization of projects are as follows:

1. 87 Hardware Projects
2. 7 Hardware/Software Projects
3. 8 Software Projects

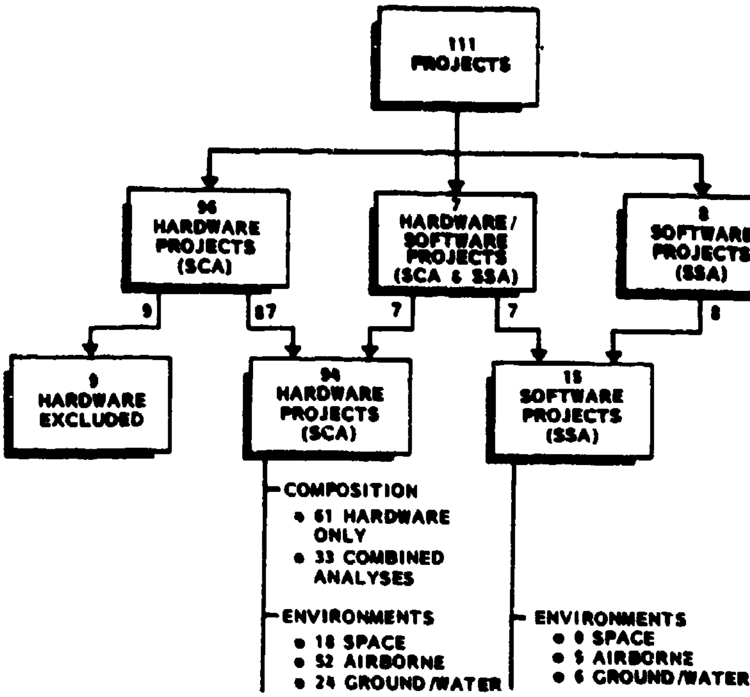


Figure 3-2. Sneak Analysis Project Summary

Because of the small number of software projects, each of the combined hardware/software projects was split and counted as a single hardware project and a single software project. The number of projects was then revised to 109 and redistributed and recategorized as follows:

1. 94 Hardware Projects
2. 15 Software Projects

The composition of the 94 hardware projects was broken down one step further and categorized as follows:

1. 61 projects were for hardware SCA only
2. 33 projects involved a blending of hardware SCA with additional analyses These additional analyses involve one or more of the following:
    a. Change Analysis
    b. Procedure Analysis
    c. Mission Support and Analysis
    d. Preliminary Hazard Analysis
    e. Fault Tree Analysis
    f. Failure Mode and Effects Analysis and Criticality Analysis
    g. Common Cause Failure Analysis
    h. Power and Load Analysis
    i. Worst Case Analysis
    j. Accident Analysis

k. Grounding Analysis
l. Fault Hazard Analysis
m. Mean-Time-Between-Failure Analysis
n. Load Switching Analysis

The composition of the 15 software projects can be broken down into two categories, as follows:

a. 13 Assembly Language Projects
b. 2 Combined Assembly Language and High Order Language (HOL) Projects

On the basis of the 109 separate hardware and software projects, 86% of the projects are hardware related, while 14% of the projects are software related. In general, the beginning of the hardware Sneak Analysis capability can be traced to the 1967 time period, and software Sneak Analysis can be traced to the 1973 time period. Hardware technique development and limited usage of the technique can be considered to have begun in 1967. Software technique development began in 1973 and was further developed in study contracts in the 1975 and 1976 time period. Software analysis application projects began in 1977. Thus, the number of software Project History Table entries is low because the technique implementation has its origin in the 1977 time period, a ten year lag behind the hardware technique.

The Project History Table entries are categorized by the following three Mission Environments and grouped by hardware and software, as follows:

1. Space Environment
   a. 18 Hardware Projects
   b. 0 Software Projects
2. Airborne Environment
   a. 52 Hardware Projects
   b. 9 Software Projects
3. Ground/Water Environment
   a. 24 Hardware Projects
   b. 6 Software Projects

The number of projects performed in any one environment is highest in Airborne, followed by Ground/Water and lowest in Space. While the number of actual projects performed in the Space Environment is lowest, this environment accounts for the longest average project duration and the largest average project funding.

The following subsections of Section 3.2 contain a summarized collection of tabular data. Some of the tables represent a particular data arrangement of the entire Appendix A Project Tables, while others are separate tables for the Space, Airborne and Ground/Water Environments. Numerous tables have been omitted from this report because no clear trends were identified in the data. This approach provides an in-depth insight into the relevant Project History data. The major areas of study are:

1. Sneak Analysis Project Phasing
2. Sneak Analysis Project Costing
3. Program Costing
4. Sneak Analysis Project Equipment

3.2.1 Sneak Analysis project phasing. A detailed study of Sneak Analysis project phasing has been performed which examined various parameters in relation to the overall program development phases. Each Sneak Analysis project was categorized into one of the following program development phases:

1. Conceptual (CON)
2. Validation (VALID)
3. Full Scale Engineering Development (FSED)
4. Full Scale Prototype Development (FSPD)
5. Pilot Production (PP)
6. Unlimited Production (UNP)

Detailed tabular data are presented for the following item comparisons:

1. Phasing/Number of Sneak Analysis Projects
2. Phasing/Number of Sneak Analysis Reports
3. Phasing/Equipment Type
4. Phasing/Equipment Criticality
5. Phasing/Equipment Classification

3.2.1.1 Phasing/number of Sneak Analysis projects: With the exception of one Ground/Water Environment project, all Sneak Analysis projects have occurred in the last four program development phases, as shown in Tables 3-1, 3-2 and 3-3.

The tables illustrate the distribution of hardware and software Sneak Analysis projects keyed to initiation of the analysis in the various program development phases. Hardware in this context refers to a hardware Sneak Analysis, which may include additional hardware analyses and may also be combined with a software analysis.

The largest number of projects are hardware types, accounting for 86% of the samples, while software occurs in 14% of the samples. No software projects have been performed for the Space Environment nor for two program phases of the Ground/Water Environment. Virtually no hardware or software projects have been performed in the conceptual or validation phases. Sneak Analysis could have utility in support of PDR at the validation phase, but no projects have been undertaken to do so. Performing the analysis earlier in the development cycle should save program dollars since changes increase in cost as the program matures.

As a composite, the ranking of projects by phase is as follows:

1.	Unlimited Production	- 35%
2.	Full Scale Engineering Development	- 26%
3.	Full Scale Prototype Development	- 21%
4.	Pilot Production	- 17%
5.	Validation	- 1%
6.	Conceptual	- 0%

TABLE 3-1 PROJECT PHASING/PROJECT TYPE, SPACE ENVIRONMENT

PROJECT TYPE \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE	0	0	6	1	4	7	18
SOFTWARE	0	0	0	0	0	0	0
TOTALS	0	0	6	1	4	7	18

TABLE 3-2 PROJECT PHASING/PROJECT TYPE, AIRBORNE ENVIRONMENT

PROJECT TYPE \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE	0	0	8	19	5	20	52
SOFTWARE	0	0	1	2	4	2	9
TOTALS	0	0	9	21	9	22	61

TABLE 3-3 PROJECT PHASING/PROJECT TYPE, GROUND/WATER ENVIRONMENT

PROJECT TYPE \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE	0	1	9	1	3	10	24
SOFTWARE	0	0	4	0	2	0	6
TOTALS	0	1	13	1	5	10	30

As a composite, the ranking of projects by Environment is as follows:

1. Airborne - 55%
2. Ground/Water - 28%
3. Space - 17%

The projects occurring at FSED were generally undertaken to support program CDR, and are most notable (percentage wise) in the Space and Ground/Water Environments. The projects occurring at FSPD are predominantly for the Airborne Environment. This program phase supports the equipment/software assembly effort prior to DSARC 2, approval for operational test and evaluation. Pilot Production projects are equally distributed between the three environments on a percentage basis. This phase supports the DSARC III milestone, which is approval for full-scale production. The last program phase is Unlimited Production and represents the largest occurrence of Sneak Analysis projects. Many of these projects were undertaken to identify problems encountered in fielded systems or when modifications were made to fielded systems.

3.2.1.2 Phasing/number of Sneak Analysis reports: The following tables contain a cross-section of hardware and software Sneak Analyses reports by program development phases. The hardware report types are:

a. Sneak Circuit Reports (SCR's)
b. Design Concern Reports (DCR's)
c. Drawing Error Reports (DER's)

The software report types are:

a. Software Sneak Reports (SSR's)
b. Software Design Concern Reports (SDCR's)
c. Software Drawing Error Reports (SDER's)

The SCR's and SSR's are conditions identified in the system (hardware/software) which will inhibit the occurrence of a desired function or will generate the occurrence of an undesired function, without regard to equipment or software failure.

The DCR's and SDCR's are conditions identified in the system which could affect performance and reliability, or where undesirable design practices have been found.

The DER's and SDER's are conditions identified within the documentation (e.g., schematics, wire lists, procedures, listings) supplied by the manufacturing contractors or agencies.

Table 3-4 represents the composite environment report averages of all Sneak Analysis projects, while Tables 3-5, 3-6, and 3-7 represent the report averages for Space, Airborne and Ground/Water Environments.

TABLE 3-4. PROJECT PHASING/REPORT AVERAGES, COMPOSITE

SNEAK ANALYSIS REPORTS \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	AVERAGES
HARDWARE							
SCR'S	0	0	36	11	10	7	15
DCR'S	0	0	17	12	11	8	11
DER'S	0	0	140	28	26	22	52
SUBTOTALS	0	0	193	51	47	37	79
SOFTWARE							
SSR'S	0	0	3	3	18	7	10
SDCR'S	0	0	12	9	23	12	16
SDER'S	0	0	4	13	37	24	21
SUBTOTALS	0	0	19	25	78	43	47

Table 3-4 illustrates a definite trend in average number of hardware reports released by program phases. The trend indicates that the earlier in the program development cycle, the greater the number of reports released. This same trend is also present within each of the three report types. This appears to be an expected result. In the FSED phase, the design is primarily a paper design, with little hardware equipment. Design oversights and problems with merging of various technologies into meaningful systems and functions occur at this phase. The FSPD phase involves the fabrication and limited subsystem testing which eliminates some of the more obvious equipment problems. Pilot production brings all of the systems together for a complete article which can be used for operational evaluation. At this phase, many of the true sneak conditions emerge that were not detected and corrected in the design phases. The Unlimited Production Phase shows the overall lowest average number of reports. This should also be an expected result, since many of the sneak conditions should have been found in prior reviews, tests, and evaluations. There are, however, still a significant number of reports released in this phase which identify conditions embedded in the equipment. Additional reviews and more extensive testing may bring the number of conditions down, but there appears to be a threshold level of conditions which are not adaptable to or identifiable by other analyses.

Table 3-4 illustrates a different trend for software report averages, possibly due to the limited number of projects. The curve rises slowly in the FSED and FSPD phases, peaks in the Pilot Production Phase, and declines slowly in the Unlimited Production Phase. Since software Sneak Analysis has been used primarily with completed software system code, the predominant phases of project

performance are understandably in the Pilot and Unlimited Production Phases. Detailed design, but little or no complete software code, is available at FSED. FSPD results in some code development, primarily in the form of modules or subroutines, however, no composite code is available. The interconnecting module linkages are still design concepts, and are implemented in the latter stage of this phase. When the entire software code is integrated together at the Pilot Production Phase for test, evaluation, and operational usage (equivalent to the latter stages of verification and validation), the greatest number of software problems are identified. This also includes the Unlimited Production Phase, where field experience problems and system modifications are the principal areas of the analysis.

Table 3-4 is built from actual project data but it may provide a misleading trend. The trend apparently indicates a greater problem finding capability in the latter development phases when the cost to the procuring activity for correcting problems is highest. This is a fundamental error in approach. In a well organized development plan, analyses should be scheduled early enough in the program development cycle to identify and correct design and operating problems in a cost effective manner. The trend results, however, are heavily skewed by actual results, which theoretically will change as additional projects are performed in Sneak Analysis. The most important element which will cause this software report trend to change will be the acceptance and use of software Sneak Analysis in software verification and validation efforts. Analysis will then be directed to detailed system design and discrete program modules in earlier development phases. Analysis can then be performed on sections of code and design before the complete software package is available. The anticipated problem report levels of software Sneak Analysis in the earlier development phases should then be roughly equivalent to those for hardware Sneak Analysis. Table 3-4 would then contain similar trends for hardware and software projects. The high software report averages in the last two phases may also indicate a higher number of reportable conditions for software than for hardware.

Tables 3-5 through 3-7 illustrate the project Sneak Analysis Report averages for the Space, Airborne and Ground/Water Environments. These tables show (individually) some differences from the trends identified in Table 3-4.

TABLE 3-5. PROJECT PHASING/REPORT AVERAGES, SPACE ENVIRONMENT

SNEAK ANALYSIS REPORTS \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	AVERAGES
HARDWARE							
SCR'S	0	0	99	55	20	7	43
DCR'S	0	0	13	0	21	8	14
DER'S	0	0	431	67	60	14	166
SUBTOTALS	0	0	548	122	101	29	223

The Space Environment appears to follow the composite trends in Table 3-4 most closely. This environment also illustrates the highest averages per project; this is due to the fact that three of the FSED projects are Apollo/ASTP, Skylab and Shuttle, which are long duration and high funding projects. Even with these projects removed, the overall trend of a declining number of reports by phase and beginning at FSED is still present. A minor "glitch" occurs at FSPD since this contains a single project result, and represents an early analysis project when emphasis was on Sneak Circuit Reports, not on Design Concern Reports. No software projects were performed in the Space Environment.

TABLE 3-6. PROJECT PHASING/REPORT AVERAGES, AIRBORNE ENVIRONMENT

SNEAK ANALYSIS REPORTS \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	AVERAGES
HARDWARE							
SCR'S	0	0	20	8	6	5	9
DCR'S	0	0	15	12	5	7	10
DER'S	0	0	45	27	12	22	26
SUBTOTALS	0	0	80	47	23	34	45
SOFTWARE							
SSR'S	0	0	2	3	17	7	10
SDCR'S	0	0	8	9	20	12	14
SDER'S	0	0	2	13	32	24	23
SUBTOTALS	0	0	12	25	69	43	47

TABLE 3-7. PROJECT PHASING/REPORT AVERAGES, GROUND/WATER ENVIRONMENT

SNEAK ANALYSIS REPORTS \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	AVERAGES
HARDWARE							
SCR'S	0	0	8	14	4	11	9
DCR'S	0	0	18	13	8	9	12
DER'S	0	0	31	18	6	26	24
SUBTOTALS	0	0	57	45	18	46	45
SOFTWARE							
SSR'S	0	0	4	0	21	0	10
SDCR'S	0	0	13	0	30	0	18
SDER'S	0	0	5	0	48	0	19
SUBTOTALS	0	0	21	0	99	0	47

Tables 3-6 and 3-7 show similar trends for the hardware and software projects. The overall number and types of reports are virtually identical for hardware and are similar for software. The only deviations occur in hardware for two distinct phases, most notably in the Pilot Production Phase and then in the FSED Phase. The hardware report levels in the Pilot Production Phase are somewhat below the composite trendline and may be due to the limited number of projects (12) in this category. The Ground/Water Environment, while following the overall trend, is low in the average number of SCR's released.

The software curve is predominantly influenced by the Airborne Environment projects, and appears to be repeated in the Ground/Water Environment, where only two phases are represented.

The averages (mean) for Tables 3-4 through 3-7 are presented in the right-hand position of each table. Standard deviations and variances within and across phases apparently show no correlation because of the range of data.

3.2.1.3 Phasing/equipment type: Each of the Sneak Analysis projects has been categorized by one or more equipment type names. The equipment type names describe the type of hardware and/or software that are predominant in the Sneak Analysis project. Selecting these names is more a function of the specific subsystems analyzed than that of the overall program. For example, the system analyzed may be an Automated Flight Control System with a high digital composition placed inside a vintage airplane that is predominantly analog or relay driven. The categorization for this project would then be digital, even though the airplane circuitry is predominantly relay logic. Up to six equipment type names could be applied to describe any single project. As such, the multiple typing of some projects results in a total of 167 equipment type names.

Table 3-8 illustrates the equipment composition of all 109 Sneak Analysis projects.

TABLE 3-8 PROJECT PHASING/EQUIPMENT TYPE, COMPOSITE

EQUIPMENT TYPE \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
RELAY	0	1	14	8	6	27	56
DIGITAL	0	0	14	17	10	13	54
ANALOG	0	1	7	10	7	7	32
MICROPROCESSOR	0	0	1	4	1	2	8
ASSEMBLY	0	0	5	2	6	2	15
HOL	0	0	1	0	1	0	2
TOTALS	0	2	42	41	31	51	167

Relay logic and digital logic systems account for 66% of the equipment composition for Sneak Analysis projects shown in Table 3-8. The Sneak Analysis technique was developed originally to handle relay logic, which in many ways is similar to the two state logic of digital equipment. Analog equipment accounts for 19% of the overall totals, followed by assembly languages at 9% and microprocessors and High Order Languages for the remaining 6%.

This table does not illustrate a definite trend and appears to indicate that equipment types analyzed as a function of development phase are not correlatable. Overall totals in Table 3-8 indicate a relatively flat or equal occurrence of equipment types by phase. However, the Space Environment has a higher average rate of combined equipment projects (1.8), than either the Airborne Environment (1.6) of the Ground/Water Environment (1.4). The overall average is 1.6 equipment types per project.

The overall table is composed of approximately 50% single equipment type projects and 50% multiple equipment type projects. By eliminating development phase and substituting time in yearly increments beginning with 1967, it can be shown that a majority of single equipment analysis projects occurred in the early historical phase (prior to 1976), while the more recent projects (1976 to present) have been blends of two or more equipment types.

3.2.1.4 Phasing/equipment criticality: The following tables contain a cross-section of hardware and software criticality rankings by program development phases. The descriptions of criticality are basically:

a. Criticality I - Loss of Life
b. Criticality II - Loss of Mission
c. Criticality III - Mission Degradation

An additional criticality level of rank 4 was included in the original criteria, but based on an analysis of the projects, no entries were made and this category was removed from the tables.

TABLE 3-9. PROJECT PHASING/CRITICALITY RANKING, SPACE ENVIRONMENT

CRITICALITY \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE							
I	0	0	3	0	0	3	6
II	0	0	3	1	4	4	12
III	0	0	0	0	0	0	0
SUBTOTALS	0	0	6	1	4	7	18

TABLE 3-10. PROJECT PHASING/CRITICALITY RANKING, AIRBORNE ENVIRONMENT

CRITICALITY \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE							
I	0	0	2	6	0	8	16
II	0	0	4	9	2	7	22
III	0	0	2	4	3	5	14
SUBTOTALS	0	0	8	19	5	20	52
SOFTWARE							
I	0	0	0	1	1	0	2
II	0	0	0	0	1	2	3
III	0	0	1	1	2	0	4
SUBTOTALS	0	0	1	2	4	2	9
TOTALS	0	0	9	21	9	22	61

TABLE 3-11. PROJECT PHASING/CRITICALITY RANKING, GROUND/WATER ENVIRONMENT

CRITICALITY \ DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE							
I	0	0	2	1	1	3	7
II	0	1	4	0	2	5	12
III	0	0	3	0	0	2	5
SUBTOTALS	0	1	9	1	3	10	24
SOFTWARE							
I	0	0	0	0	2	0	2
II	0	0	3	0	0	0	3
III	0	0	1	0	0	0	1
SUBTOTALS	0	0	4	0	2	0	6
TOTALS	0	1	13	1	5	10	30

The majority of Sneak Analysis projects has been confined to the man or mission critical subsystems, Criticalities I and II, respectively. For hardware systems, 80% of the projects are in these two categories, while software systems occupy 67% of these categories.

The Airborne Environment Table 3-10 accounts for approximately half of the Sneak Analysis projects and shows pronounced peaks in the hardware FSPD and Unlimited Production Phases. The distribution of Airborne rankings by Program Phase is highest for Criticality II projects, as is the case with the remaining tables.

Software shows no definable trends and contains too small a sample to analyze in these tables.

An additional analysis of equipment composition is provided in Table 3-12, which represents the ranking of equipment classification to program phase for the composite environments. The predominant program phases are the Full-Scale Engineering Development and Unlimited Production phases. The predominant equipment classification is the Control system for both hardware and software. The Command classification is the next most populous category, since it includes a high user interface and contains documented procedure checklists. The Communication classification is the lowest category of Sneak Analysis applications. The relative ranking of the categories is: Control (60%), Command (30%), and Communication (10%).

TABLE 3-12 PROJECT PHASING/EQUIPMENT CLASSIFICATION, COMPOSITE

EQUIPMENT CLASSIFICATION / DEVELOPMENT PHASE	CONCEPT	VALIDATION	FSED	FSPD	PILOT PRODUCTION	UNLIMITED PRODUCTION	TOTALS
HARDWARE							
COMMAND	0	1	11	8	6	18	44
CONTROL	0	1	23	21	12	36	93
COMMUNICATION	0	1	8	5	2	5	21
SUBTOTALS	0	3	42	34	20	59	158
SOFTWARE							
COMMAND	0	0	4	1	3	1	9
CONTROL	0	0	5	2	6	2	15
COMMUNICATION	0	0	1	1	0	0	2
SUBTOTALS	0	0	10	4	9	3	26
TOTALS	0	3	52	38	29	62	184

The Space Environment results for the Sneak Analysis projects are the most balanced of the three environments, with a relatively high occurrence of projects in the Communication category. Several of the projects in this sample involved analysis of spacecraft telemetry systems. Approximately half of the projects occur in the Control category, which is the lowest percentage level of the three environments.

The Airborne Environment results illustrate the highest average Control category level, with approximately two-thirds of the samples in this category. The Communication category results are the lowest of the entire sample, even considering that the Airborne Environment represents the largest population of the three environments.

The Ground/Water Environment contains the highest occurrence of the Command category in the three environments. The Ground/Water Environment is composed of many projects which included airborne and missile ground test equipment, nuclear systems, and drilling equipment. This equipment was designed for extensive operator interaction. This factor plus the performance of Failure Mode and Effects Analyses, Fault Tree Analyses, and Common Cause Failure Analyses contributed to the high level (one-third of the samples) for the Command category.

3.2.2 Sneak Analysis project costing. The Appendix A Project History Tables were analyzed for Sneak Analysis cost patterns in relation to various parameters. The parameters desired were:

1. Sneak Analysis Cost/Program Cost
2. Sneak Analysis Cost/Development Phase
3. Sneak Analysis Cost/Period of Performance
4. Sneak Analysis Cost/Equipment Type
5. Sneak Analysis Cost/Equipment Criticality
6. Sneak Analysis Cost/Equipment Classification
7. Sneak Analysis Cost/Number of Reports

The category distributions of Sneak Analysis dollar costs were selected in $100,000 increments to provide a meaningful sample. The costs are raw costs, unadjusted for inflation. Approximately 75% of the overall Sneak Analysis costs are under the $100,000 level.

3.2.2.1 Sneak Analysis cost/program cost: Table 3-13 was compiled to determine the relationship between program development cost and Sneak Analysis cost.

Table 3-13 illustrates 75% of Sneak Analysis projects are under $100,000, 89% under $200,000, and 95% under $300,000. One-third of the hardware projects are for programs under $100 million, and one-third are for programs over $1 billion. The remaining one-third of the hardware projects are for programs between $100 million and $1 billion, with a majority of these programs under $500 million. 40% of the software projects are for programs under $100 million and 27% are for programs over $1 billion. The remaining one-third of the software projects are for programs between $100 million and $1 billion, virtually the same trend as the hardware projects.

TABLE 3-13. SNEAK ANALYSIS COST/PROGRAM COST, COMPOSITE

PROJECT COST (M) \ SNEAK ANALYSIS COST (K)	0 - 100K	100K - 200	200 - 300	300 - 400	>400	TOTALS
HARDWARE						
0 - 100 M	31	1	0	0	0	32
100 - 200	7	1	0	1	0	9
200 - 300	3	0	0	0	0	3
300 - 400	2	1	1	0	0	4
400 - 500	1	1	0	1	0	3
500 - 600	0	1	0	0	0	1
600 - 700	0	1	0	0	0	1
700 - 800	0	0	0	0	0	0
800 - 900	0	0	0	0	0	0
900 - 1B	3	2	0	0	0	5
1B+	23	6	3	0	4	36
SUBTOTALS	70	14	4	2	4	94
SOFTWARE						
0 - 100 M	5	1	0	0	0	6
100 - 200	3	0	0	0	0	3
200 - 300	0	0	0	0	0	0
300 - 400	0	0	0	0	0	0
400 - 500	0	0	1	0	0	1
500 - 600	0	0	0	0	0	0
600 - 700	0	0	0	0	0	0
700 - 800	0	0	0	0	0	0
800 - 900	0	0	0	0	0	0
900 - 1B	1	0	0	0	0	1
1B+	3	0	1	0	0	4
SUBTOTALS	12	1	2	0	0	15
TOTALS	82	15	6	2	4	109

A greater percentage (50%) of Space Environment hardware Sneak Analysis projects occur with costs greater than $100,000. The majority of projects in excess of $100,000 is in this environment and represents the Apollo/ASTP, Skylab and Shuttle projects. The program cost levels under $100 million and greater than $1 billion account for the majority of hardware Sneak Analysis projects.

Airborne Environment hardware projects are more polarized than the other environments, with 31% of the projects under the $100 million level and 46% over the $1 billion level. Software projects exhibit the same polarization.

Ground/Water Environment hardware projects illustrate the same polarization, but a reversal of the endpoints with the majority of the projects (46%) in the $100 million range and 25% in the $1 billion range. Software projects are limited to under $200 million programs.

3.2.2.2 Sneak Analysis cost/development phase: Table 3-14 was compiled to determine the relationship between program development phases and the distribution of Sneak Analysis costs.

TABLE 3-14. SNEAK ANALYSIS COST/DEVELOPMENT PHASING, COMPOSITE

SNEAK ANALYSIS COST (K) / DEVELOPMENT PHASE	0 - 100	100 - 200	200 - 300	300 - 400	>400	TOTALS
HARDWARE						
CONCEPT	0	0	0	0	0	0
VALID	1	0	0	0	0	1
FSED	18	1	0	1	3	23
FSPD	16	3	1	0	1	21
PP	9	1	2	0	0	12
UNP	26	9	1	1	0	37
SUB TOTALS	70	14	4	2	4	94
SOFTWARE						
CONCEPT	0	0	0	0	0	0
VALID	0	0	0	0	0	0
FSED	4	1	0	0	0	5
FSPD	2	0	0	0	0	2
PP	5	0	1	0	0	6
UNP	1	0	1	0	0	2
SUB TOTALS	12	1	2	0	0	15
TOTALS	82	15	6	2	4	109

Table 3-14 illustrates the same basic project distribution trends as presented in Section 3.2.2.1. The largest development phase is Unlimited Production, for 36% of the projects, while the largest Sneak Analysis cost category is under $100,000, which occurs for 75% of the projects.

By using this table as a base and summing the overall project costs by development phase, an average cost for Sneak Analysis projects was determined. The figures used are raw dollar entries and include all projects, including the three large Space Environment projects. Table 3-15 illustrates the average project costs per development phase. The double entry numbers appearing in the last two columns of the table are averages for projects with and without the three large Space Environment projects.

The overall project cost is $163,000, while the hardware average is $177,000, and the software average is $75,000. However, the three large Space Environment projects greatly influence these averages. Removing these three entries from the tables results in an overall project cost of $75,000, with hardware and software projects at equal $75,000 levels. In Section 3.2 it was noted that one-third of the hardware projects were blended tasks involving Sneak Analysis and one or more related analyses. Since the cost averages shown include the entire hardware analysis set, the true cost for a hardware Sneak Analysis project is less than $75,000.

TABLE 3-15. SNEAK ANALYSIS AVERAGE COST/DEVELOPMENT PHASE, COMPOSITE

SNEAK ANALYSIS COST (K) / DEVELOPMENT PHASE	0 - 100	100 - 200	200 - 300	300 - 400	>400	TOTALS
HARDWARE						
CONCEPT	0	0	0	0	0	0
VALID	72	0	0	0	0	72
FSED	34	160	0	391	3277/0	478/88
FSPD	52	136	256	0	426	94
PP	40	131	279	0	0	84
UNP	32	139	206	367	0	71
SUB TOTALS	39	138	255	379	2564/0	177/75
SOFTWARE						
CONCEPT	0	0	0	0	0	0
VALID	0	0	0	0	0	0
FSED	19	200	0	0	0	55
FSPD	16	0	0	0	0	16
PP	43	0	300	0	0	86
UNP	41	0	255	0	0	148
SUB TOTALS	30	200	278	0	0	75
TOTALS	37	142	262	379	2564/0	163/75

The following distribution of costs within hardware projects was noted:

1. $ 0 - $100,000, average $ 39,000, Deviation ± $26,000
2. $100,000 - $200,000, average $138,000, Deviation ± $22,000
3. $200,000 - $300,000, average $255,000, Deviation ± $36,000
4. $300,000 - $400,000, average $379,000, Deviation ± $12,000
5. $400,000 - Up , average $2,564,000, Deviation ± $1,810,000

The following distribution of costs within software projects was noted and subject to a limited number of projects:

1. $ 0 - $100,000, average $ 39,000, Deviation ± $21,000
2. $100,000 - $200,000, average $200,000, Single Project
3. $200,000 - $300,000, average $278,000, Two Projects
4. $300,000 - Up , No Projects

Standard deviation calculations were performed for each development phase in the tables, but no real correlation existed.

3.2.2.3 Sneak Analysis cost/period of performance: The period of performance (number of calendar months for Sneak Analysis) was examined in relation to the project cost to determine relevant trends, if any. Table 3-16 provides the distribution of Sneak Analysis projects as a function of period of performance.

TABLE 3-16. SNEAK ANALYSIS COST/PERIOD OF PERFORMANCE, COMPOSITE

PERIOD PERFORMANCE (MONTHS) \ SNEAK ANALYSIS COST (K)	0 - 100	100 - 200	200 - 300	300 - 400	> 400	TOTALS
HARDWARE						
0-3	19	0	0	0	0	19
3-6	33	3	0	0	0	36
6-9	11	5	1	1	0	18
9-12	5	1	0	1	1	8
> 12	2	4	4	0	3	13
SUB TOTALS	70	13	5	2	4	94
SOFTWARE						
0-3	3	0	0	0	0	3
3-6	6	0	0	0	0	6
6-9	3	1	1	0	0	5
9-12	0	0	0	0	0	0
> 12	0	0	1	0	0	1
SUB TOTALS	12	1	2	0	0	15
TOTALS	82	14	7	2	4	109

From an overall hardware project perspective, the following distributions were determined for Sneak Analysis project duration:

1. 0 - 3 Months, 20% of Projects
2. 3 - 6 Months, 38% of Projects
3. 6 - 9 Months, 19% of Projects
4. 9 - 12 Months, 9% of Projects
5. > 12 Months, 14% of Projects

Three-fourths of the projects were under nine months in duration and approximately 60% of the projects were under six months in duration.

From an overall software project perspective, the following distributions were determined for Sneak Analysis project duration:

1. 0 - 3 Months, 20%
2. 3 - 6 Months, 40%
3. 6 - 9 Months, 33%
4. 9 - 12 Months, 0%
5. > 12 Months, 7%

The trend for software projects appears to correlate with the hardware and is, in fact, heavily influenced by the method of dividing combined hardware/software Sneak Analysis tasks referred to in Section 3.2.

The Space Environment has the highest average project duration of 20 months, the Airborne Environment project duration is 6.5 months, and the Ground/Water Environment project duration is 6 months.

3.2.2.4 Sneak Analysis cost/equipment type: Table 3-17 was compiled to determine the relationship between Sneak Analysis cost and type of equipment included in the analysis.

TABLE 3-17. SNEAK ANALYSIS COST/EQUIPMENT TYPE, COMPOSITE

EQUIPMENT TYPE \ SNEAK ANALYSIS COST (K)	0 - 100K	100 - 200	200 - 300	300 - 400	>400	TOTALS
RELAY	40	9	3	1	3	56
DIGITAL	44	5	4	0	2	55
ANALOG	24	1	3	1	2	31
MICROPROCESSOR	5	0	1	0	2	8
ASSEMBLY	12	0	3	0	0	15
HIGH ORDER	0	0	2	0	0	2
TOTALS	125	15	16	2	9	167

A meaningful analysis of these tables cannot be performed because the relative percentages of project dollars to the equipment type categories were not maintained in the historical files and will only be known for the ongoing Sneak Analysis projects. Instead, the tables can only be used to show the category distributions.

Individual distributions of hardware equipment types by environment closely approximate the Table 3-17 hardware entries. Software projects are heavily distributed in the under $100,000 category.

3.2.2.5 Sneak Analysis cost/equipment criticality: Table 3-18 illustrates the distribution of Sneak Analysis projects by criticality ranking and by cost categories.

The Criticality I hardware projects in excess of $100,000 occurred for 30% of the projects; Criticality II hardware projects in excess of $100,000 occurred for 29% of the projects; and, Criticality III hardware projects in excess of $100,000 occurred for only 11% of the projects.

TABLE 3-18. SNEAK ANALYSIS COST/CRITICALITY RANKING, COMPOSITE

SNEAK ANALYSIS COST (K) / CRITICALITY	0 - 100K	100 - 200	200 - 300	300 - 400	> 400	AVERAGES
HARDWARE						
I	21	4	1	0	4	30
II	32	8	3	2	0	45
III	17	2	0	0	0	19
SUB TOTALS	70	1[illegible]	4	2	4	94
SOFTWARE						
I	3	0	1	0	0	4
II	5	0	1	0	0	6
III	4	1	0	0	0	5
SUB TOTALS	12	1	2	0	0	1[illegible]
TOTALS	82	15	6	2	4	109

Overall, Criticality I hardware projects resulted in an average project cost of $402,000; however, this again included the three large Space Environment projects. When these three projects were removed, the overall Criticality I hardware cost average was $70,000. Overall, Criticality II hardware cost average was $88,000, while Criticality III was $48,000.

Overall, Criticality I software projects resulted in an average project cost of $97,000; Criticality II had an average project cost of $68,000; and Criticality III had an average project cost of $65,000.

3.2.2.6 Sneak Analysis cost/equipment classification: Table 3-19 represents the composite data for the equipment classification as a function of Sneak Analysis cost. The distribution of each of the three equipment classifications is uniform throughout the various hardware cost categories. The software distribution is too limited to establish any trends based on Sneak Analysis cost. The predominant occurrence of projects for both hardware and software is the Control category in the 0-$100,000 cost range.

The portion of Table 3-19 which includes the Space Environment results represents the most uniform distribution of equipment classifications and Sneak Analysis costs. The three large projects, including Apollo, Skylab and Shuttle, are equally distributed for equipment classification within the greater than $400K cost category. The three projects raise the overall Space Environment numerical levels for the Command and Communication categories and moderate the typically high level of the Control category. Although difficult to substantiate, the trend toward more equal distributions within equipment classification is an anticipated consequence for the higher cost (>$400K) Sneak Analysis projects. Depth of analysis, number of components, interconnectivity of systems, higher criticality of systems, and possibly unresolved problems are primary factors for this more uniform distribution.

TABLE 3-19. SNEAK ANALYSIS COST/EQUIPMENT CLASSIFICATION, COMPOSITE

SNEAK ANALYSIS COST (K) / EQUIPMENT CLASSIFICATION	0 - 100	100 - 200	200 - 300	300 - 400	>400	TOTALS
HARDWARE						
COMMAND	33	5	2	1	3	44
CONTROL	69	14	4	2	4	93
COMMUNICATION	15	2	1	0	3	21
SUBTOTALS	117	21	7	3	10	158
SOFTWARE						
COMMAND	6	2	1	0	0	9
CONTROL	12	2	1	0	0	15
COMMUNICATION	2	0	0	0	0	2
SUBTOTALS	20	4	2	0	0	26
TOTALS	137	25	9	3	10	184

3.2.2.7 Sneak Analysis cost/number of reports: Table 3-20 represents the averaged composite data for the number of Sneak Analysis Reports issued to the procuring activity as a function of Sneak Analysis cost.

TABLE 3-20. SNEAK ANALYSIS COST/REPORT AVERAGES, COMPOSITE

SNEAK ANALYSIS COST (K) / SNEAK ANALYSIS REPORTS	0 - 100	100 - 200	200 - 300	300 - 400	>400	AVERAGES
HARDWARE						
SCR's	7.9	10.6	23	18.5	148	15.2
DCR's	9.3	9.2	34.3	12.5	29.3	11.3
DER's	18.3	25.9	115.5	126.5	639.8	52.3
SUBTOTALS	35.5	45.7	172.8	157.5	817.1	78.8
SOFTWARE						
SSR's	7.7	9	22.5	-0-	-0-	9.7
SDCR's	12	32	31.5	-0-	-0-	15.9
SDER's	15.3	1	67	-0-	-0-	21.3
SUBTOTALS	35	42	121	-0-	-0-	46.9
REPORT AVERAGES	35.5	45.5	155.5	157.5	817.1	74.4

The hardware trend shows an increasing number of reports generated as a function of increased Sneak Analysis cost. Only in the $300-400K range does the average deviate from the apparent trend, and this is probably due to the average being based on only two projects. In general, the increasing average from category to category is attributable both to the size of the system and the greater number of subsystems included in the Sneak Analysis project. The complexity and interconnectivity of systems represent two of the major causes of designed-in Sneak conditions. The more a system function crosses system interfaces, the more likely the occurrence of reportable problems.

The software trend also shows an apparently increasing number of reports as a function of increased Sneak Analysis cost. However, there are a significant number of projects in only the first category, $0-100K. The software trend is thus inconclusive because of the low number of samples in the greater than $100K project cost ranges.

Drawing Error Reports (DER's and SDER's) constitute a large portion of each sample; this is typical for a Sneak Analysis project. However, the increasing trend for hardware and software Sneak Reports (SCR's, DCR's, SSR's, SDCR's) is still present. An overall project average is 26 reports (SCR's, DCR's, SSR's, SDCR's) for both hardware and software systems.

Table 3-21 contains the Space Environment results obtained by comparing Sneak Analysis Reports to Sneak Analysis Cost. No software projects have been performed for the Space Environment, which is the reason abbreviated tables are presented.

TABLE 3-21. SNEAK ANALYSIS COST/REPORT AVERAGES, SPACE ENVIRONMENT

SNEAK ANALYSIS COST (K) / SNEAK ANALYSIS REPORTS	0 - 100	100 - 200	200 - 300	300 - 400	> 400	AVERAGES
HARDWARE						
SCR's	4.8	17	58	-0-	197	42.8
DCR's	5.3	11.4	44	-0-	32.6	13.7
DER's	18.2	30.8	134	-0-	845	165.9
REPORT AVERAGES	28.3	59.2	236	-0-	1074.6	222.8

The Space Environment Table 3-21 illustrates the smoothest trend of the three environments in each of the categories. The Space Environment also shows the highest overall hardware averages in the sample. The average number of reports increases in all cases, except in the $300-400K range (no data) and the low DCR report average in the greater than $400K category. The low DCR average is due to the lack of emphasis originally associated with identifying design concerns in the Apollo and Skylab projects. In these early projects, the primary reports were the Sneak Circuit Reports (SCR's) and Drawing Error Reports (DER's).

Table 3-22 contains the Airborne Environment results obtained by comparing Sneak Analysis Reports to Sneak Analysis Cost.

TABLE 3-22. SNEAK ANALYSIS COST/REPORT AVERAGES, AIRBORNE ENVIRONMENT

SNEAK ANALYSIS COST (K) / SNEAK ANALYSIS REPORTS	0 - 100	100 - 200	200 - 300	300 - 400	> 400	AVERAGES
HARDWARE						
SCR's	9.1	5.3	11.3	-0-	1	8.6
DCR's	8.9	7.3	31	-0-	19	10.1
DER's	21.3	19.4	109.3	-0-	24	26.2
SUBTOTALS	39.3	32	151.6	-0-	44	44.9
SOFTWARE						
SSR's	10	-0-	9	-0-	-0-	9.9
SDCR's	14.6	-0-	13	-0-	-0-	14.4
SDER's	19.9	-0-	46	-0-	-0-	22.8
SUBTOTALS	44.5	-0-	68	-0-	-0-	47.1
REPORT AVERAGES	40.2	32	130.8	-0-	44	45.2

The Airborne Environment Table 3-22 does not indicate a clear trend in the hardware report averages. The number of projects covered in the hardware portion of this table is 52, more than a significant number of samples, but highly concentrated in the $0-100K and $100-200K ranges. The last two category ranges are insignificant and are not considered typical.

The Airborne Environment hardware averages are the highest of the samples in the $0-100K range, even considering the Space Environment averages. The averages compare favorably with those in the Ground/Water Environment, which are shown in Table 3-23. The averages in the $100-200K range, however, are lower than anticipated. The report averages may be low due to the fact that several of the hardware projects in this category were combined analyses, and no adjustment of project dollar values was made.

All of the Airborne Environment software projects, with the exception of one, are in the $0-100K range. No trends can be identified with this distribution. It is significant to note, however, that the report averages in this range are noticeably higher than the corresponding range in the Ground/Water Environment.

Table 3-23 contains the Ground/Water Environment results obtained by comparing Sneak Analysis Reports to Sneak Analysis Cost.

TABLE 3-23. SNEAK ANALYSIS COST/REPORT AVERAGES, GROUND/WATER ENVIRONMENT

SNEAK ANALYSIS COST (K) / SNEAK ANALYSIS REPORTS	0 - 100	100 - 200	200 - 300	300 - 400	> 400	AVERAGES
HARDWARE						
SCR's	6.9	13	-0-	18.5	-0-	8.4
DCR's	12	10.5	-0-	12.5	-0-	11.9
DER's	12.2	36.5	-0-	126.5	-0-	23.8
SUBTOTALS	31.1	60	-0-	157.5	-0-	44.0
SOFTWARE						
SSR's	3	9	36	-0-	-0-	9.5
SDCR's	6.8	32	50	-0-	-0-	18.2
SDER's	6.3	1	88	-0-	-0-	19.0
SUBTOTALS	16.1	42	174	-0-	-0-	46.7
REPORT AVERAGES	28.6	54	174	157.5	-0-	44.6

The hardware report averages in the Ground/Water Environment Table 3-23 show the same increasing trend although no samples are present for the $200-300K and greater than $400K ranges. The number of Sneak Reports (SCR's and DCR's) compares favorably to the corresponding range in the Space Environment.

The software report averages exhibit a similar trend as in the hardware, but are comprised of a limited sample size.

3.2.3 Program costing. The analysis of the number of Sneak Analysis Reports to the overall program costs resulted in no significant trends being identified. Program cost appears to have little if any predictable effect on the number of reports generated. Distribution of program cost into 11 separate dollar ranges may be spreading the 109 project results too thinly. Variations of large and small Sneak Analysis projects, program phase differences, and equipment differences appear to significantly influence the averages within any dollar range.

Table 3-24 represents the averaged composite data for the number of Sneak Analysis Reports as a function of overall program costs.

TABLE 3-24. PROGRAM COST/REPORT AVERAGES, COMPOSITE

SNEAK ANALYSIS REPORTS \ PROGRAM COST (m)	0 - 100	100 - 200	200 - 300	300 - 400	400 - 500	500 - 600	600 - 700	700 - 800	800 - 900	900 - 1,000	> 1B	AVERAGES
HARDWARE												
SCR's	7.1	13.8	1.3	29.8	10	-0-	21	-0-	-0-	4.6	24.1	15.2
DCR's	11.2	13.1	5	15.3	7	-0-	8	-0-	-0-	12.8	11.3	11.3
DER's	19.2	34.8	8.7	18.5	50.3	-0-	63	-0-	-0-	17.2	92.8	52.3
SUBTOTALS	37.5	61.7	15.0	63.5	67.3	-0-	92	-0-	-0-	34.6	128.2	78.8
SOFTWARE												
SSR's	10.2	7.3	-0-	-0-	36	-0-	-0-	-0-	-0-	5	5.5	9.7
SDCR's	18.8	6.0	-0-	-0-	50	-0-	-0-	-0-	-0-	10	12	15.9
SDER's	24.3	6.7	-0-	-0-	88	-0-	-0-	-0-	-0-	2	15.8	21.3
SUBTOTALS	53.3	20.0	-0-	-0-	174	-0-	-0-	-0-	-0-	17.0	33.3	46.9
REPORT AVERAGES	40.0	51.3	15.0	63.5	88.6	-0-	92	-0-	-0-	31.7	118.7	74.4

3.2.4 Sneak Analysis project equipment. The Appendix A Project History Tables were analyzed for equipment selection patterns in relation to various parameters. Equipment descriptions in terms of equipment composition and complexity, that is type, have been presented, to some extent, in the preceding Project Phasing and Project Costing sections. Additional detailed tabular data are presented for the following item comparisons:

1. Type of Equipment or Software/Program Environment
2. Type of Equipment or Software/Criticality Ranking
3. Equipment Criticality/Equipment Classification
4. Equipment Criticality/Sneak Analysis Reports

3.2.4.1 Equipment type/environment: Table 3-25 illustrates the distribution of equipment type categories by Project Environment:

TABLE 3-25. EQUIPMENT TYPE/ENVIRONMENT

ENVIRONMENT \ EQUIPMENT TYPE	RELAY	DIGITAL	ANALOG	MICRO-PROCESSOR	ASSEMBLY	HIGH ORDER LANGUAGE	TOTALS
SPACE	13	12	6	1	0	0	32
AIRBORNE	27	32	19	6	9	0	93
GROUND/WATER	16	11	6	1	6	2	42
TOTALS	56	55	31	8	15	2	167

The relay and digital equipment categories are by far the main areas included in Sneak Analysis projects, with a majority of projects performed for airborne applications. However, the key item identified is that the total of 167 equipment occurrences were for 109 projects. Many of the analyses were performed for a combination of equipment types, and this appears to be the case for all recent analyses. No particular predominant pairing or grouping of equipment types were identified.

3.2.4.2 Equipment type/criticality: Table 3-26 illustrates the distribution of Sneak Analysis projects by equipment type and criticality ranking.

TABLE 3-26. EQUIPMENT TYPE/CRITICALITY, COMPOSITE

CRITICALITY \ EQUIPMENT TYPE	RELAY	DIGITAL	ANALOG	MICRO-PROCESSOR	ASSEMBLY	HIGH ORDER LANGUAGE	TOTALS
I	18	15	11	4	4	1	53
II	30	28	15	2	6	0	81
III	8	11	6	2	5	1	33
TOTALS	56	54	32	8	15	2	167

While the numerical level of equipment type totals is highest for Criticality II, the overall percentage is virtually equal for the Criticality I and II projects and lowest for Criticality III projects. The percentage is computed by dividing the Table 3-26 entries by the number of projects in each of the criticality categories.

3.2.4.3 Equipment criticality/equipment classification: Table 3-27 presents the composite environment results by correlating equipment classification to the equipment criticality ranking. Criticality II projects which involve mission critical systems represent the largest table sample, both for hardware and software. Criticality I systems are more uniformly distributed between the three equipment classifications. Criticality II and III systems are concentrated more toward the Control category.

3.2.4.4 Equipment criticality/Sneak Analysis Reports: Table 3-28 represents the averaged composite data and reflects the number of Sneak Analysis Reports as a function of equipment or software criticality ratings. The trends identified are coarse indicators of Sneak Analysis project effectiveness based on the criticality of the subsystems analyzed.

The hardware and software report average trends show the prominence of Criticality I systems, followed by Criticality II systems, and ending in the lowest average level in Criticality III systems.

TABLE 3-27. CRITICALITY RANKING/EQUIPMENT CLASSIFICATION, COMPOSITE

EQUIPMENT CLASSIFICATION \ CRITICALITY	I	II	III	TOTALS
HARDWARE				
COMMAND	17	18	9	44
CONTROL	29	46	18	93
COMMUNICATION	10	11	0	21
SUBTOTALS	56	75	27	158
SOFTWARE				
COMMAND	3	4	2	9
CONTROL	4	6	5	15
COMMUNICATION	1	1	0	2
SUBTOTALS	8	11	7	26
TOTALS	64	86	34	184

TABLE 3-28. CRITICALITY RANKING/REPORT AVERAGES, COMPOSITE

SNEAK ANALYSIS REPORTS \ CRITICALITY	I	II	III
HARDWARE			
SCR's	28.2	10.1	6.6
DCR's	12.6	9.6	13.1
DER's	105.5	29.9	21.4
SUBTOTALS	146.3	49.6	41.2
SOFTWARE			
SSR's	16.3	9.5	4.8
SDCR's	20.8	16.0	12.0
SDER's	31.3	31.2	2.6
SUBTOTALS	68.3	56.7	19.4
REPORT AVERAGES	137.1	50.3	36.6

Hardware exhibits the highest report averages in Criticality I systems. The hardware trend is still prevalent when consideration is given to only the average number of SCR's and DCR's. The one disruption to the trend is the average number of DCR's in Criticality III. Radars, radios, transmitters, and other monitoring equipment are the main systems included in the Criticality III category. These systems apparently have a higher incidence of design incompatibilities than the other categories. All other trends indicate an increasing report average the higher the system criticality (Criticality I - Loss of Life, II - Loss of Mission, III - Mission Degradation).

The software trend illustrates a smooth increase in reports by increasing criticality rating, with no exceptions. Fifteen projects constitute the total sample for the software table.

Table 3-29 contains the Space Environment results obtained by comparing the Sneak Analysis Reports to Criticality ratings. The results are greatly influenced by the three large projects (Apollo, Skylab, and Shuttle) and tend to elevate Criticality I systems to a high threshold value. This is the predominant factor in the high level of Criticality I hardware projects in Table 3-28. The report averages for Criticality II systems compare favorably with the corresponding categories of the other two environments.

TABLE 3-29. CRITICALITY RANKING/REPORT AVERAGES, SPACE ENVIRONMENT

SNEAK ANALYSIS REPORTS \ CRITICALITY	I	II	III
HARDWARE			
SCR's	98.6	15.4	-0-
DCR's	21.6	9.8	-0-
DER's	425.3	36.3	-0-
SUBTOTALS	545.5	61.5	-0-

Table 3-30 contains the Airborne Environment results obtained by comparing the Sneak Analysis Reports to Criticality ratings.

The differences in average number of hardware reports by Criticality rating are less pronounced in the Airborne Environment. The overall trend of increased number of reports for increased criticality systems still prevails, however. Criticality II and III results are virtually equal.

Software exhibits an unusual pattern in that Criticality II systems are the most prominent and significantly so. The number of SDER's is the major influence to this trend, but the number of SSR's and SDCR's is also significantly high. The peak is produced by one large software project involving an airborne weapon control system.

TABLE 3-30. CRITICALITY RANKING/REPORT AVERAGES, AIRBORNE ENVIRONMENT

SNEAK ANALYSIS REPORTS \ CRITICALITY	I	II	III
HARDWARE			
SCR's	10.9	7.4	7.5
DCR's	10.5	10.1	9.6
DER's	30.6	24.1	23.9
SUBTOTALS	52.0	41.7	41.0
SOFTWARE			
SSR's	11.5	17.0	3.8
SDCR's	12.0	26.0	7.0
SDER's	15.0	54.3	3.0
SUBTOTALS	38.5	97.3	13.8
REPORT AVERAGES	50.6	48.7	34.9

Table 3-31 contains the Ground/Water Environment results obtained by comparing Sneak Analysis Reports to Criticality Ratings.

TABLE 3-31. CRITICALITY RANKING/REPORT AVERAGES, GROUND/WATER ENVIRONMENT

SNEAK ANALYSIS REPORTS \ CRITICALITY	I	II	III
HARDWARE			
SCR's	9.6	9.4	4.2
DCR's	10.0	8.5	22.8
DER's	13.1	33.8	14.6
SUBTOTALS	32.7	51.7	41.6
SOFTWARE			
SSR's	21.0	2.0	9.0
SDCR's	29.5	6.0	32.0
SDER's	47.5	6.0	1.0
SUBTOTALS	98.0	14.0	42.0
REPORT AVERAGES	47.2	44.1	41.7

The Criticality I hardware report averages in Table 3-31 are low due to the majority of projects occurring as blended analyses. In one project, the predominant analysis effort is a Failure Mode and Effects Analysis. The remainder of the hardware averages compare favorably across environments.

The software averages are based on a very small number of projects. The trend appears to be similar to the composite system trends, but due to the small sample size, the results are inconclusive.

3.3 Task 3 - Comparison and Description of Analysis Techniques.

Task 3 - Investigate and determine the similarities or dissimilarities of a Sneak Circuit Analysis to other types of analyses such as: failure modes, effects, and criticality analysis; wiring and schematic drawing reviews; and fault tree. Areas of overlapping coverage shall be defined and the analysis that is most effective in correcting deficiencies in those areas shall be identified.

Each of the analysis techniques in this effort has been described and included in a comparison matrix. These analyses have been implemented along with Sneak Analysis.

Hardware analysis techniques are presented first, followed by software analysis techniques.

3.3.1 Hardware analysis technique descriptions. An engineering analysis is an examination of the nature of a system by examining, in detail, the design of the system's parts. The examination is always considered in the context of the system's environment. In fact, the assumed environment is the only element that differs in some of these analyses. A detailed description is presented for the following analysis techniques:

a. Sneak Analysis
b. Failure Modes and Effects Analysis
c. Fault Tree Analysis
d. Common Cause Failure Analysis
e. Sensitivity Analysis
f. Worst Case Analysis
g. Power and Load Analysis
h. Grounding Analysis
i. Accident Analysis
j. Hazard Analysis
k. Preliminary Hazard Analysis
l. Operations Hazard Analysis
m. Fault Hazard Analysis

3.3.1.1 Sneak Analysis: Sneak Analysis examines system operations during normal conditions for design oversights. It consists of two subanalyses: Sneak Circuit Analysis for electrical-electronic systems, and Software Sneak Analysis for computer programs. Historically, sneak conditions have escaped rigid design screens, resulting in program schedule delays, damage to equipment during test, and increased downtime during operation. Program cost effectiveness may, therefore, be increased by utilizing Sneak Analysis to reduce life cycle costs.

The application of Sneak Analysis to operational systems provides a means of evaluating that the system will operate as designed.

The Boeing Company initiated development of the Sneak Circuit Analysis technique in 1967 in response to the National Aeronautics and Space Administration's concern for crew safety in manned spacecraft operations. NASA surmised that crew safety was endangered by electrical problems due to latent paths in the electrical design.

These paths, called "sneaks," are hidden in the electrical circuitry and exhibit unapparent cause-effect relationships, which may inhibit desired operations or initiate unintended actions, without being contingent upon component failure. The Sneak Analysis technique was formally extended to identify sneak conditions in software in 1973.

3.3.1.2 Sneak Circuit Analysis: The data used for Sneak Circuit Analysis must represent the system circuitry as it actually is or will be constructed, contingent upon quality control checks, tests and inspections. All reports are written against these drawings. Analysis based on the detailed circuit drawings identifies more system conditions than an analysis performed on system or functional level schematics. The higher level drawings frequently represent design intent or a perception of intended system design. The process of translating this design into detailed schematics and wire lists typically results in latent circuit conditions. For this reason, analysis at the higher level involves a risk that not all of the problems will be found.

In early program development phases, detailed drawings are not available and the system level drawings of necessity must be used for the analysis. Problems will be identified at this higher level, but the analysis should be extended to later design phases so that the system configuration can be analyzed in detail.

Sneak Circuit Analysis is a unique approach to discovery of latent conditions which cause unwanted functions to occur or which inhibit wanted functions, independent of component failure. The technique involves accumulation of detailed circuit diagrams and wire lists, arrangement of circuit elements into topological network trees, and examination of these network trees for suspected sneak circuits (reference Figure 3-3).

Direct analysis of manufacturing and installation schematics is difficult as these documents are laid out to facilitate hookup by technicians without regard to circuit function. So many details and unapparent continuities exist in these drawings that an analyst could become entangled and lost in the maze. The first task of the sneak circuit analyst is to convert this detailed, accurate information into a form usable for analytical work. In many cases, the magnitude of data manipulation required for this conversion necessitates the use of computer automation. In projects having a small data base, it has been found that manual data manipulation could be employed. In either case, the detailed schematics are converted into topological network trees, drawn so that electrical current (power) is considered to flow down the page.

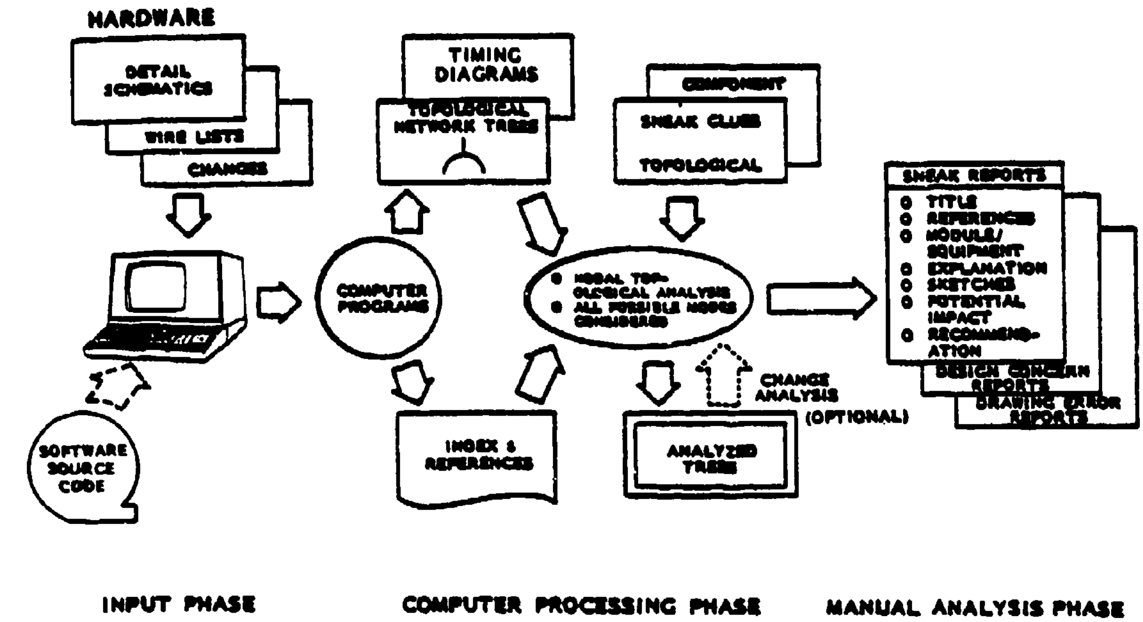


Figure 3-3. Sneak Circuit Analysis Task Flow

Once the trees have been produced, the next task of the analyst is to identify the basic topological patterns that appear in each tree. Five basic patterns exist: the Single Line (No-Node) Topograph, the Ground Dome, the Power Dome, the Combination Dome, and the "H" Pattern (as shown in Figure 3-4 below; "PWR" represents electrical power, "S" indicates a switching element, and "L" indicates an electrical load). The "H" pattern typically has the highest incidence of problems due primarily to the higher number of power sources, returns, loads and switches. The main problem occurs with the "H" crossbar, which includes L3, S3 and S4. This can result in power reversals, ground reversals, and current reversals.

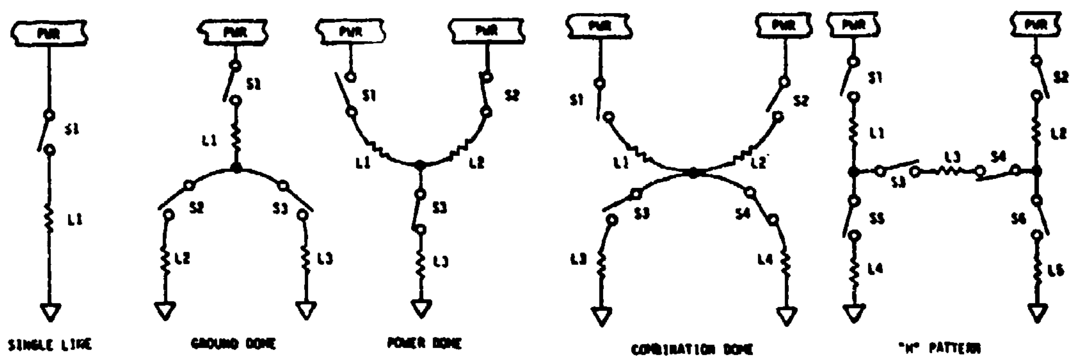


Figure 3-4. Basic Topographs

Although at first glance, a given circuit may appear more complex than these basic patterns, closer inspection reveals that the circuit is actually composed of these basic patterns in combination. As the sneak circuit analyst examines each node in the network tree, he must identify which pattern or patterns best describe the node. The analyst then applies the basic clues that have been found to typify sneak circuits involving that particular pattern. These clues represent questions that the analyst must answer about the interrelationships of circuit elements involved in the pattern. The questions are systematically formatted to lead to the identification of any capability of the circuit to experience a surprise or sneak condition at the node being analyzed. Off-nominal modes are considered equally with normal operations, and no assessment of probabilities is attempted in standard Sneak Circuit Analysis.

The clues have been compiled over a 15-year period and have been refined and updated to accommodate new equipment technologies. The developed clues are typically proprietary to the performing contractor. A very basic clue in the Power Dome, Combination Dome, and "H" Pattern Dome is the reversal of the two power sources. In some systems, the two power sources of the "H" pattern are to be mutually exclusive, and the lower circuitry must provide proper isolation. If isolation is not maintained, a bus to bus sneak is generated. Two equal power sources can still generate sneaks, whenever one bus develops an increased or decreased voltage level relative to the second bus. The resultant voltage and current shifts can inadvertently activate components in the "H" pattern. A short on one bus could short the second bus, resulting in inducing undesired equipment functions and no convenient means or capability to reset the system.

The sneak circuits are classified into four basic types:

a. Sneak Paths - which cause current or energy to flow along an unexpected route.

b. Sneak Timing - which may cause or prevent the flow of current or energy to activate or inhibit a function at an unexpected time.

c. Sneak Indications - which may cause an ambiguous or false display of system operating conditions.

d. Sneak Labels - which may cause incorrect stimuli to be initiated through operator error.

When a potential sneak condition is identified, the analyst must verify that it is valid. The circuit is checked against the latest applicable drawings or revisions, and operational information may be reviewed concerning the system in question. If the sneak condition is verified, a Sneak Circuit Report is written which includes applicable drawings, an explanation of the condition(s), system level impact, and a recommendation for elimination of the sneak.

During the course of analysis, unnecessary or undesirable circuit conditions are sometimes encountered. Such conditions as certain single failure points, unsuppressed inductive loads, unnecessary components, and inadequate redundancy provisions are reported in the Design Concern Reports.

Obvious inconsistencies between source drawings or differences between the source drawings and other descriptive documentation that may be supplied are called drawing errors. While most drawing errors are uncovered in data encoding or tree drawing phases, some become apparent while investigating suspected sneak circuits. These data discrepancies are documented by Drawing Error Reports.

3.3.1.3 Failure Modes and Effects Analysis (FMEA): A Failure Modes and Effects Analysis is a causal analysis. Each component of the system may be considered, dependent on the desired depth of analysis. Each possible failure mode of the considered component is treated as a separate case. For each case, the analyst determines the effect of the failure, the function the component provides, the method by which the failure can be detected, and the criticality of the case. The process is basically a "bottom-up" analysis, where the analyst postulates a knowl equipment failure and tries to identify all system effects given a particular system configuration. The process of this analysis is recorded in a form headed as shown below.

FAILURE MODES AND EFFECTS ANALYSIS WORKSHEET

COMPONENT IDENTIFICATION	FUNCTION	FAILURE MODE	FAILURE EFFECT	METHOD OF DETECTION	CRITICALITY

Component failure modes are dependent on several factors. A partial list of failure modes is shown below. Note that all components cannot have all of these failure modes, and some modes may not be considered on a particular project:

a. Structural Failure (Rupture)
b. Physical Binding/Jamming
c. Fails To Remain (In Position)
d. Fails To Open
e. Fails To Close
f. Fails Open
g. Fails Closed
h. Internal Leakage
i. External Leakage
j. Fails Out Of Tolerance
k. Inadvertent Operation
l. Intermittent Operation
m. Erratic Operation
n. Erroneous Indication
o. Restricted Flow
p. Fails To Stop
q. Fails To Start
r. Fails To Switch
s. Premature Operation
t. Delayed Operation
u. Erroneous Output (Reduced)
v. Loss Of Output (Thrust, Indication, Partial, False, Etc.)
w. Shorted
x. Open (Electrical)
y. Leakage (Electrical)

The effects of the component failure must, ultimately, be considered in terms of the operation of the system. However, the analyst's first task is to determine the system configuration, through which the faults will be propagated. Certain categories of system effects can be defined. This is useful where the effects impact a subsystem only. The following is a partial list of these categories:

a. No Effect
b. Loss of Redundancy
c. Functional Degradation
d. Subsystem Degradation
e. Loss of Function
f. Loss of Subsystem
g. Loss of Interface Redundancy
h. Degradation of Interface Function
i. Degradation of Interface Subsystem
j. Loss of Interface Function
k. Loss of Interface Subsystem

The criticality of a failure effect reflects the danger to the system or personnel. Three levels of criticality are usually sufficient. These criticalities are:

a. Criticality I - damage to personnel or destruction of the system.

b. Criticality II - causes the system to cease operation or fail, wherein, the next associated failure would cause damage to personnel or destruction of the system.

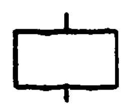
c. Criticality III - degrades system operation.

The Method of Detection column is filled in to describe how the system operator can detect that the failure has occurred. Detection may generally be accomplished by either the operation of a control panel indicator or by failure of the system to perform as expected.

3.3.1.4 Fault Tree Analysis: Fault Tree Analysis is a "top-down" analysis that is basically deductive in nature. The analyst identifies failure paths by use of a fault tree drawing. A fault tree is a graphical representation of a thought process. It is constructed from events and logical operators. An event is either a component failure or system operation. The events and their graphical representation are shown below:

<u>EVENT REPRESENTATIONS</u>

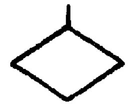
The rectangle identifies an event that results from the combination of fault events through the input logic gate.



The rectangle is also used to describe a conditional input to an INHIBIT GATE. It indicates a condition that is presumed to exist for the life of the system.

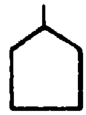
The circle describes a basic fault event that requires no further development. Frequency and mode of failure of items so identified are derived from empirical data.

The diamond describes a fault event that is considered basic in a given fault tree. The possible causes of the event are not developed, either because the event is of insufficient consequences or the necessary information is unavailable.

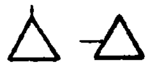


The oval is used to record the conditional input to an INHIBIT GATE. It defines the state of the system that permits a fault sequence to occur, and may be either normal to the system or result from failures.

The house indicates an event that is normally expected to occur such as a phase change in a dynamic system.



The triangles are used as transfer symbols. A line from the apex of the triangle indicates a "transfer in" and a line from the side denotes a "transfer out."



The double diamond is used in the simplification of a fault tree for numerical evaluation. The event describes results from the causes that have been identified but are not shown on a particular version of the fault tree.

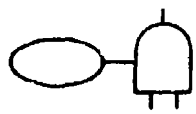
Events are connected by logic operations that describe Boolean functions. The logic operations are shown below.

LOGIC OPERATIONS

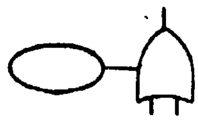
AND GATE describes the logical operation whereby the coexistence of all input events is required to produce the output event.

OR GATE defines the situation whereby the output event will exist if one or more of the input events exists.

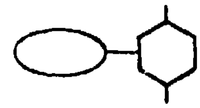
PRIORITY AND GATE performs the same logic function as the AND GATE with the additional stipulation that sequence as well as coexistence is required.

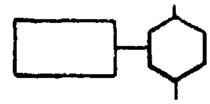


EXCLUSIVE OR GATE functions as an OR GATE with the restriction that specified inputs cannot coexist.



INHIBIT GATES describe a causal relationship between one fault and another. The input event directly produces the output event if the indicated condition is satisfied. The conditional input defines a state of the system that permits the fault sequence to occur, and may be either normal to the system or result from failures. It is represented by an oval if it describes a specific failure mode and a rectangle if it describes a condition that may exist for the life of the system.





A fault tree is begun by selecting a top event. This event is the ultimate disaster or undesired event. From there, the analyst endeavors to find the immediate events that can, in some logical combination, cause the top event. These lower events are examined, in turn, for causes and the process is repeated to levels of greater detail. Ideally, the lowest level events will be all basic events and represented by a circle. This is not always the case, however, and many diamonds may be found at the bottom of the tree.

A fault tree provides a method for determining the logical causes of a given event. It illustrates all of the ways an undesired event can occur. It helps determine the critical components and the need for other analytical efforts. Numerical computations indicating the probability of occurrence for the top event and intermediate events can be obtained. The major drawback of the fault tree is that there is no way to insure that all causes have been evaluated consistently.

The Fault Tree Analysis is also performed on the system configuration, determined by the analyst. Determining the configuration of a system is generally central to all analyses.

3.3.1.5 Common Cause Failure Analysis (CCFA): A common cause event is defined as a single event or condition that results in multiple component failures. This analysis starts by determining the top events of interest. Critical sets are then determined. A critical set is a group of components that control a function, the failure of which would cause the top event to occur. Critical sets are most easily determined from a fault tree.

Next, the commonality of the critical sets is determined. Commonality is defined as elements common to a number of components of the critical set. Examples of commonality are shared connectors, common location, wire bundles and cooling elements. A partial list of commonalities will give some idea of the areas that could be covered:

a. Energy Source
b. Calibration
c. Installation Contractor
d. Maintenance
e. Operator or Operation
f. Proximity
g. Test Procedure
h. Energy Flow Paths

The credible failure modes of the components of the critical set are determined. These failure modes are then linked to causes which could result in multiple component failure. A list of commonly encountered causes is given below:

a. Impact
b. Vibration
c. Pressure
d. Grit
e. Moisture
f. Stress
g. Temperature
h. Electromagnetic Interference (EMI)
i. Radiation Damage
j. Conducting Medium
k. Out-of-Tolerance Voltage
l. Out-of-Tolerance Current
m. Corrosion
n. Other Chemical Reaction
o. Carbonization
p. Biological

Finally, the analyst determines the system level result of the failures. Again, the analyst needs a system configuration to determine the effects of the fault(s) propagation. The analyst also determines the methods of recovering from the failures' effects. Actions taken by the analyst are recorded on a worksheet with the heading shown below:

CCFA WORKSHEET

CRITICAL FUNCTION SET	COMMONALITY	CRITICAL EVENT	POTENTIAL CAUSE	EFFECT	REMARKS

3.3.1.6 Sensitivity analysis: Since the solution to a design problem is not unique, different networks can be constructed to produce the same input impedance or transfer function. As long as ideal elements are used under ideal conditions, one network works just as well as the other. In practice, however, one network may outperform another because it is less sensitive to element variations and to environmental changes. This network may be no more expensive to construct than the other. A sensitivity analysis is a parametric analysis that determines how a system is affected as the values of its constituent components varies from the nominal. Note that these variations do not exceed the manufacturer's stated tolerance for the part.

In order to perform this analysis, a quantitative measure is needed to compute networks with regard to element variations from the ideal. Sensitivity functions are used for this purpose. These functions provide a numerical measure of how much an important aspect of the system or response varies as an element, or a combination of elements varies from the nominal (design) values.

During a sensitivity analysis, the analyst must first determine the type of circuits represented in the system. The analyst then obtains the sensitivity function for each circuit type, makes the calculations, and determines the effects each circuit variation has on the system output. Components that have the greatest sensitivity, with respect to the system output, are placed on a critical parts list so that they may receive special attention. The analyst should also recommend equivalent networks with lower sensitivity, where possible.

3.3.1.7 Worst case analysis: A variation of sensitivity analysis is the worst case analysis. In this analysis the values of all the components are considered to be at the manufacturer's limit of acceptance. The resulting system change or effect is given by the sum of the individual changes. The worst case occurs when all changes are either positive or negative. The circuit output is next checked to determine 1) if it results in an undesirable system output, and 2) if some system components will be stressed beyond their tolerance.

3.3.1.8 Power and load analysis: A power and load analysis consists of determining open circuit voltage and closed or short circuit current on lines that can control hazardous functions. The open circuit case determines what voltage is available in relation to other lines in wire bundles and connectors. The current sourcing capabilities of a line are determined to detect whether any critical components are being overstressed.

Current is evaluated by first determining the steady state short circuit current for shorts occurring at various susceptible points. Next, a power profile is made for each potential short. This is a plot of short circuit current versus time. In order to calculate the current, path resistance must be determined using wiring dimensions, specifications, and routing information so that the analyst has a system configuration on which to base the analysis.

Transient currents are calculated from the type cabling, the types of switched loads on the line, and the presence of RF emitters. EMI sources must be fully considered when they can be identified.

3.3.1.9 Grounding analysis: In equipments that are not in direct electrical contact with the earth, such as, vehicles and other isolated systems, improper electrical grounding can cause a problem. The analyst must search schematics for conditions which would allow EMF differences to develop between various points of the ground tree. In order to make this determination, the analyst must use data that represents the system as it actually is or will be constructed.

The first task in the grounding analysis is to create a current tree of the ground wires. This tree is similar to those used in Sneak Circuit Analysis projects, where each node and connection can be shown. It is particularly important that the tree indicates where the ground changes structures (e.g., in a train it is the car-to-car ground wire). The tree is examined to determine which topological structure it contains. Then, clues are applied to the tree to determine if current flow can occur. Current flow determination is reported as a sneak circuit.

3.3.1.10 Accident analysis: The analysis of the effects of accidents can be divided into the categories of crash and fire. Crash type accidents can result in shorted connector pins, severing wires which can cause momentary shorting between wires and permanent open circuits, and the loss of protective components such as capacitors used to shunt high frequencies to ground. Fire damage can cause electrical shorts and discontinuities in wire bundles, as well as, the change of component characteristics due to elevated temperatures.

The techniques of power and load analysis and common cause failure analysis are used to evaluate the effects of accidents on critical circuitry. Previous accident data, if available, should be used to identify particularly susceptible equipment and determine the probability of damage to critical circuitry. The routing and mounting of wire bundles and the location and mounting of connectors must be considered in this type of analysis. Consideration must also be given to the susceptibility of the system enclosure to crushing.

3.3.1.11 Hazard Analyses: Hazard analyses are performed to identify hazardous conditions. They should consider the system, hardware, software, facility, personnel, and their interrelationships during the complete life of the system. Hazard analyses are primarily used to determine a measure of system safety. The types of hazard analyses that will be discussed here are Preliminary Hazard Analysis, Operational Hazard Analysis, and Fault Hazard Analysis.

3.3.1.12 Preliminary Hazard Analysis: The Preliminary Hazard Analysis is an examination of the generic hazards known to be associated to a system at its conceptual phase of development. The purpose of this analysis is to:

a. Identify hazards
b. Determine the effects of the hazards
c. Establish initial safety requirements
d. Determine areas to monitor for safety problems
e. Initiate the planning of a safety program
f. Establish safety scheduling priority
g. Identify areas for testing
h. Identify the need for additional analyses

The Preliminary Hazard Analysis determines the recognized and anticipated design safety pitfalls and provides the method by which these pitfalls may be avoided. When this analysis is undertaken, there is little information on design details and less on procedures. The Preliminary Hazard Analysis is usually a top-level review for safety problems. In most instances, the following basic steps are undertaken for a Preliminary Hazard Analysis:

a. Review problems known through past experience on similar products or systems to determine whether they could also be present in the equipment under development.

b. Review the mission and basic performance requirements, including the environments in which operations will take place.

c. Determine the primary hazards that could cause injury, damage, loss of function, or loss of material.

d. Determine the contributory and initiating hazards that could cause or contribute to the primary hazards listed.

e. Review possible means of eliminating or controlling the hazards, compatible with mission requirements.

f. Analyze the best methods of restricting damage in case there is a loss of control of a hazard.

g. Indicate who is to take corrective action, and the actions that each will accomplish.

Three basic approaches that can be used to insure that all hazards are covered are the columnar form, top level fault tree, and narrative description. These methods will not in themselves find hazards. They will orient the analyst so that a thorough coverage of all aspects of the system will be performed.

The columnar form is the simplest methodology to implement. The chief advantage is that it is easy to review. The form has a heading that patterns questions in the mind of the analyst. The headings must at least be as shown as follows:

PRELIMINARY HAZARD ANALYSIS WORKSHEET

HAZARD	CAUSE	EFFECT	HAZARD CATEGORY	CORRECTIVE OR PREVENTIVE MEASURES

The hazard is the generic area or condition that may impact system safety. The following is a partial list of hazards (the analyst can usually think of many more):

a. Acceleration
b. Contamination
b. Corrosion
d. Chemical dissociation
e. Electrical
f. Explosion
g. Fire
h. Heat and temperature
i. Leakage
j. Moisture
k. Oxidation
l. Pressure
m. Radiation
n. Chemical replacement
o. Shock (mechanical)
p. Stress concentrations
q. Stress reversals
r. Structural damage or failure
s. Toxicity
t. Vibration and noise
u. Weather and environment

The cause column is used to explain when the system is exposed to the hazard. It is here that the results of system generation are considered. Mission phasing must also be considered, as well as, an estimate of the percentage of system operation time that the hazard will be in effect.

The effect column is system centered. It details the action of the hazard on system operation. In this column the possibility of causing injury or death, however remote, must be stated.

The hazard category is a numerical measure of how important the hazard is. The number of categories should be kept small, usually four or less, so that attention may be placed where it will do the most good.

The corrective or preventive measures column is almost self-explanatory. Here, methods of abating the hazard are given.

The top level fault tree follows the method of fault tree analysis with generic events. Although this method helps define causes and effects, it does not follow that the system is checked hazard by hazard. Since the fault tree is event oriented, it helps analyze undesired events, but does not determine that a particular event is a hazardous condition, element or potential accident.

The narrative approach is less rigorous, and usually less complete, than the top level fault tree and narrative approaches. Narrative writing style is a lengthy and time consuming task. This approach is less susceptible to systematic method or technique, and therefore, the results usually have serious gaps or incomplete areas. The hazardous conditions and potential accidents are generally identified from experience, and then are explained in great depth and detail, more on the order of a final report than an analysis.

3.3.1.13 Operations Hazard Analysis: The analysis of hazards, associated with the performance of system opeations, is Operations Hazard Analysis. This analysis identifies hazardous conditions. In explaining this analysis, precise use is made of the following terms:

a. Operation - an objective to be achieved
b. Task - a basic step involved in achieving the objective
c. Procedure - method, or detailed instructions, for performing the task
d. Step - a single action performed without changing equipment

In general, operations are made up of tasks which consist of one or more procedures, which are accomplished by performing a series of steps.

In performing this analysis, one must know the configuration of the operation under analysis and have a hazard guideline. In order to identify hazards one must be able to visualize all of the internal and external system interrelationships and interdependencies. This means that the following data are necessary:

a. A description of the operation or task
b. Knowledge of the time-space relationships involved
c. Knowledge of the personnel, materials and tools being used
d. Knowledge of system and equipment design and overall operation
e. Knowledge of external factors, such as environment

Next, for each operation, the analyst identifies all the hazardous conditions. The operations are then divided into tasks, procedures and steps to identify the elements that directly cause or effect the hazard. A hazards checklist can aid the analyst in making this determination. Once the hazards are determined, methods to abate the hazards should be considered.

3.3.1.14 Fault Hazard Analysis: The purpose of this analysis is to identify the hazardous conditions and elements due to potential hardware fault conditions which could be generated within the system. It is very similar to a Failure Modes and Effects Analysis.

In doing a Fault Hazard Analysis the analyst first examines the causes and effects of component failures. Each credible failure mode of each component is first identified. Then, the causes of the component failure and the resulting effects on the system are determined. If the failure mode effect contributes to the occurrence of a potential accident, or safety critical condition, the failure mode is then revealed as a hazardous condition and the particular component as a hazardous element. The component failure rate, in conjunction with the hazard classification, indicates the relative significance of identified hazardous conditions. If a failure mode is classified as critical or catastrophic, the frequency of occurrence of the failure mode will give an indication of the risk involved. Note that this is only an indication of risk and not an absolute quantitative evaluation. Other interrelated failures could cause the probability of occurrence to increase.

The results of this analysis are recorded on a tabular form. The form is headed as shown below.

FAULT HAZARD ANALYSIS WORKSHEET

COMPONENT	FAILURE MODE	FAILURE RATE	SYSTEM MODE	FAILURE EFFECT	HAZARD CLASS	REMARKS

3.3.2 Hardware analysis technique comparison matrix. The analyses described herein complement rather than compete with each other. Table 3-32 gives a summary of the inputs, outputs and capabilities of these analyses. In many of the areas considered, there is considerable overlap. This is to be expected because the main area of disagreement is the type of output provided. As for input, all of the analyses require some form of system description. These descriptions range from block diagram (fault tree and Preliminary Hazard Analysis) to production documentation with details such as, connector pin assignments and physical locations of cable routing (grounding, accident and Common Cause Failure Analyses). Sneak Analysis occupies the middle ground requiring electrical interconnection documents, but not physical information. This commonality accounts for the overlap in the project phase comparison because various types of data only become available at certain phases in the design process.

The element comparisons are self-explanatory and will not be described further.

3.3.3 Software analysis technique descriptions. Five specific software analysis techniques are described in the material that follows. The techniques are basically static analyses. Some of the information on analysis techniques other than Software Sneak Analysis has been obtained from "Checkout Techniques, Software Reliability Guidebook," Prentice-Hall, 1979; Robert L. Glass.

3.3.3.1 Software Sneak Analysis: Data used for Software Sneak Analysis should reflect the program as it is actually written. This includes system requirements, system description, coding specifications, detailed and complete source code, a compilation listing, and operating system documentation. All reports are written against these documents.

Software Sneak Analysis is used to discover program logic which causes undesired program outputs or inhibits a desired output. The technique involves the reduction of the program source code to topological network tree representations of the program logic.

Direct analysis of program listings is difficult because the system is modular for ease of programming. Also, the code is listed as a file of records, without regard to functional flow. The first task of the software sneak analyst is to convert the program source code into a form usable for analysis. In most cases, this step requires computer conversion. In either case, the program source code is converted with reference to an input language description file into topological network trees, such that program control is considered to flow down the page. The remaining task functions are similar to those in Figure 3-3.

TABLE 3-32. HARDWARE ANALYSIS COMPARISON MATRIX

COMPARISON ELEMENT \ ANALYSIS TECHNIQUE	SNEAK CIRCUIT ANALYSIS	FAILURE MODE AND EFFECTS ANALYSIS	FAULT TREE ANALYSIS	COMMON CAUSE FAILURE ANALYSIS	SENSITIVITY ANALYSIS	WORST CASE ANALYSIS	POWER AND LOAD ANALYSIS	GROUNDING ANALYSIS	ACCIDENT ANALYSIS	PRELIMINARY HAZARD ANALYSIS	OPERATION HAZARD ANALYSIS	FAULT HAZARD ANALYSIS
PHASE												
CONCEPTUAL										X		
VALIDATION			X							X	X	
FSED	X	X	X		X	X				X	X	X
FSPD	X	X	X	X	X	X			X	X	X	X
PILOT PRODUCTION	X	X	X	X	X	X	X	X	X		X	X
UNLIMITED PRODUCTION	X	X	X	X	X	X	X	X	X		X	X
MODE												
MANUAL	X	X	X							X	X	
SEMIAUTOMATIC	X	X	X	X		X	X	X	X			X
AUTOMATIC					X							
EQUIPMENT												
RELAY	X	X	X	X			X	X	X			X
ANALOG	X	X	X	X	X	X						X
DIGITAL	X	X	X	X		X						X
ANALYSIS TYPE												
QUANTITATIVE			X	X	X	X	X	X	X			
QUALITATIVE	X	X	X	X	X	X	X	X	X	X	X	X
USE OF RESULTS												
PLANNING										X	X	
OPTIMIZATION					X							
OPERABILITY	X	X			X	X					X	
RELIABILITY	X	X			X	X						X
GRACEFUL DEGRADATION		X	X	X			X	X	X			X
PERFORMANCE UNDER STRESS		X	X	X			X	X	X	X		X
FAULT ISOLATION	X	X	X	X		X		X				
DESIGN EFFICIENCY	X				X							
DATA TYPE												
BLOCK DIAGRAM			X							X	X	
ENGINEERING DRAWINGS			X		X						X	X
INTEGRATED SCHEMATICS	X	X	X		X	X				X	X	X
PRODUCTION DRAWINGS	X	X	X	X	X	X	X	X	X		X	X
WIRE LISTS	X	X	X	X			X	X	X			
MECHANICAL DRAWINGS				X			X		X			X
CONNECTOR PIN MAPS				X			X	X	X			
PHOTOGRAPHS								X	X			
PROBLEM TYPES												
SNEAK PATH	X		X				X	X	X			
SNEAK LABEL	X										X	
SNEAK INDICATOR	X										X	
SNEAK TIMING	X				X	X						
OVER STRESSED COMPONENT	X				X	X	X		X			
WRONG COMPONENT	X				X	X	X		X			
COMPONENT FAILURE	X	X	X	X					X			X
LINE SHORT		X		X			X	X	X			
OUT OF BOUNDS TEMP				X					X	X	X	
MOISTURE LEAKAGE				X			X	X				

Once the trees have been drawn, the analyst identifies the basic topological patterns that appear in the trees. Six basic patterns exist: the Single Line, the Return Dome, the Iteration/Loop Circuit, the Parallel Line, the Entry Dome, and the Trap Circuit, as shown in Figure 3-5 below. The topological patterns containing branch or jump instructions have the highest incidence of problems. This includes the Return, Iteration and Parallel Line Domes. The crossing of module or function interfaces as a result of the branch instruction is a prime problem cause.

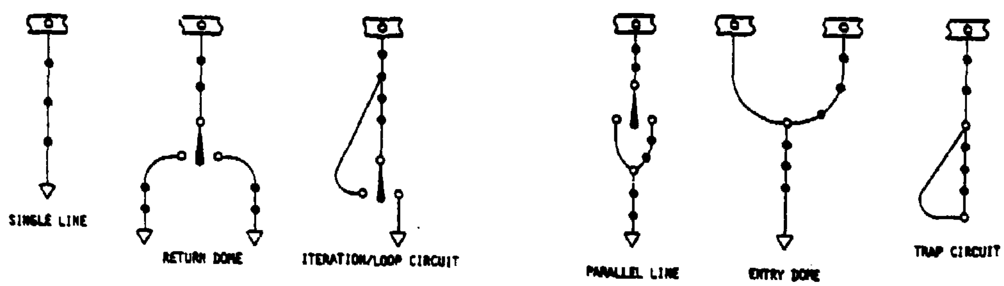


Figure 3-5. Software Topographs

Although at first glance, a given software tree may appear to be more complex than these basic patterns, closer inspection will reveal that the code is actually composed of these basic structures in combination. As each node in the tree is examined, the analyst must identify which pattern or patterns include that node. The analyst then applies the basic clues that have been found to typify the sneaks involved with that particular structure. These clues are in the form of questions that the analyst must answer about the use and interrelationships of the instructions that are elements of the structure. These questions are designed to aid in the identification of the sneak conditions in the instruction set which could produce undesired program outputs. Software clues are different than the hardware clues referenced in Section 3.5.1.2, and are typically proprietary to the performing contractor. Branch instructions can alter program flow to an incorrect location or address, encounter unitialized variables, and induce timing or sequencing problems.

Software sneaks are classified into four basic types:

a. Sneak Output - the occurrence of an undesired output.

b. Sneak Inhibit - the undesired inhibition of an input or output

c. Sneak Timing - the occurrence of an undesired output by virtue of its timing or mismatched input timing

d. Sneak Message - the program message does not adequately reflect the condition

When a potential sneak is identified, the analyst must verify that it is valid. The code is checked against the latest applicable listings and compiler information may be reviewed concerning the language in question. If the sneak is verified, a Software Sneak Report (SSR) is written which includes an explanation, system level impact, and a recommendation for elimination of the sneak.

During the course of analysis, questionable programming practice or instruction implementations are encountered. If such usage could result in program errors, it is reported in a Software Design Concern Report (SDCR).

If two or more documents, or the program listing and a supporting document, do not agree on some point, the program listing is assumed to be correct. If the analyst then determines that the condition is not a software sneak or questionable practice, a Software Documentation Error Report (SDER) is issued describing the discrepancy.

3.3.3.2 Desk checking: Desk checking is one of the earliest forms of software verification. It involves:

a. Reviewing a program listing for faults,

b. Performing arithmetic calculations to verify output value correctness, and

c. Manually simulating program execution in order to understand and verify program logic and data flow.

Since desk checking is such an ill-defined concept, it is difficult to provide a cost estimate for its use. It is undoubtedly true, however, that moderate amounts of desk checking save more money than they generate in cost.

Desk checking efforts concentrate on areas of special problems, especially suspected errors or code inefficiencies, and involve techniques appropriate to that problem.

3.3.3.3 Peer code review: A peer code review is a process by which a team of programming personnel (i.e., technologists) do an in-depth review of a program or portion of a program, by inspection. In general, the responsible programmer will verbally lead the participants sequentially through the logic flow of the program as represented in the listing. All logic branches should be taken at least once. The function of each statement will be discussed as it is encountered. Program requirements and design specifications will be present for correlation of function to its driving factors.

Peer code reviews are expensive, adding 10 to 50% to the cost of software implementation, since only about 100 source statements may be reviewed in an hour, and the concentration of the participants wanes after a short time. Thus, in many circumstances it might be wise to review key program portions and to select other portions for review randomly.

A peer code review should not occur until after coding of the program to be reviewed is completed, well annotated, and syntactically correct.

3.3.3.4 Structural Analysis: A structural analyzer is an automated tool that seeks and records errors in the structural makeup of a computer program undergoing analysis. Structural analysis is a relatively new concept, beginning in the early 1970's. Structural analyzers are almost always language and installation or project specific. Most structural analyzers built to date accommodate only FORTRAN or COBOL. For example, DAVE, built at the University of Colorado, processes CDC-6000 FORTRAN programs looking for uninitialized variables via a very elaborate algorithm.

The major factor in the cost of a Structural Analysis is the acquisition of a structural analyzer. Costs can range from trivial if the program is already in the public domain, to upwards of $100,000 for implementation of an elaborate analytical tool.

3.3.3.5 Proof of correctness: Proof of correctness is the process of using mathematical theorem-proving concepts on a computer program or its design to show that it is consistent with its specification. This is done by breaking the program into logical segments, defining input and output assertions for each segment, and demonstrating that, when the program functions, if all input assertions are true, then so too are all output assertions. It must also be shown that the program successfully terminates.

Many researchers are currently working in the proof-of-correctness area. Small algorithms and programs have been proven in this environment; however, it is at least 10 years away from being useful on programs of any significance.

Even for small simple programs, the symbolic manipulations involved in the proof of correctness technique can be overly complex, introducing errors into the computation of the statements to be proven as well as the proof of those statements. Thus, the technique would be most successful on highly mathematical and relatively straightforward segments of any program.

Lack of practical experience with proof of correctness makes it difficult to quantify costs. Usage costs are significant, possibly adding 100 to 500% to the cost of the portion of the software being proven.

3.3.4 Software Analysis Technique Comparison Matrix. The data for the Software Analysis Comparison Matrix, Table 3-33, was obtained from the following three sources:

1. "Checkout Techniques," Software Reliability Guidebook, Prentice-Hall, 1979; Robert L. Glass, pp. 86-104.

2. "Spectrum of Budgets/Costs for Software System Life Cycle Costs," Documentation of Successful Software Management Seminar, April 1981; pp. 81-812.

3. Historical data from past and present performances of Sneak Software Analysis.

TABLE 3-33. SOFTWARE ANALYSIS COMPARISON MATRIX

COMPARISON ELEMENT / ANALYSIS TECHNIQUE	SNEAK SOFTWARE ANALYSIS	DESK CHECKING	PEER CODE REVIEW	STRUCTURAL ANALYZER	PROOF OF CORRECTNESS
Type	• Program Source Listing • Design Specifications • Logic Flow Diagrams • Language Description	• Program Source • Design Specifications	• Program Source Listing • Design Specifications	• Program Source Listing • Job Control Statements	• Program Source Listing • Design Specifications
Mode	Semi-automatic	Manual	Manual	Semi-automatic	Manual
Language Type	Assembly/HOL	Assembly/HOL	Assembly/HOL	HOL (2)	Assembly/HOL
Verification & Validation Phase Application	• Requirements/Specifications Analysis • Design Analysis • Code Evaluation • Verification Testing • Validation Testing	• Code Evaluation	• Code Evaluation	• Code Evaluation • Verification Testing	• Code Evaluation
Additional Cost to Software to Perform Task	4 - 40%	25%	10 to 50%	25 to 500%	100 to 500%
	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies • Identifies Discrepancies between Program Code and Program Documentation	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies • Verifies Output Values per Specifications	• Identifies Code Inefficiencies	• Identifies Logic Flow Errors • Identifies Data Flow Errors • Identifies Code Inefficiencies	• Identifies Discrepancies between Program Code and Specifications

3.4 Task 4 - Specification and Tailoring Requirements. Task 4 - Using the data collected on sneak circuit requirements and results, develop specification, statement of work, and data item requirements that provide a capability to tailor requirements to acquisitions of various types (type of equipment, complexity, criticality, etc). Also, guidelines for monitoring a Sneak Circuit Analysis including time phasing (i.e., conceptual, advanced development, full scale development, and production phases) and schedule requirements (i.e., length of time to perform) shall be developed.

The approach for this task includes the development of descriptions and rationale for the tailoring of Sneak Analysis requirements to acquisitions of various types. The items included in this effort are:

1. Specification requirement for Sneak Analysis (Hardware and Software)

2. Request for Proposal considerations and evaluation criteria

3. Tailoring Statement of Work requests

   a. Hardware Sneak Analysis
   b. Software Sneak Analysis
   c. Integrated Hardware/Software Sneak Analysis
   d. Data Item Description
   e. Third Party (Proprietary) Data Working Agreement
   f. Project Schedule
   g. Combining Sneak Analysis with Other Analyses

The overall functional flow of the analysis selection process is shown in Figure 3-6. The Sneak Analysis specifications are currently listed in MIL-STD-785B, Reliability Program for Systems and Equipment Development and Production. The specifications in turn lead the procuring activity to consider the application guidelines developed in Task 5. If the guidelines indicate the need to incorporate Sneak Analysis in the reliability program plan, the next step would be to determine whether the acquisition is to be sole source or released for competitive bidding. In either case, the request for proposal considerations should provide insight into the items for procurement. The final step would include selection of the relevant task statement of work, data item descriptions, and possibly third party agreements for proprietary documentation.

The second portion of Task 4 also requires the development of guidelines for monitoring a Sneak Analysis task. These guidelines will aid the procuring activity in effectively monitoring and utilizing the results of Sneak Analysis.

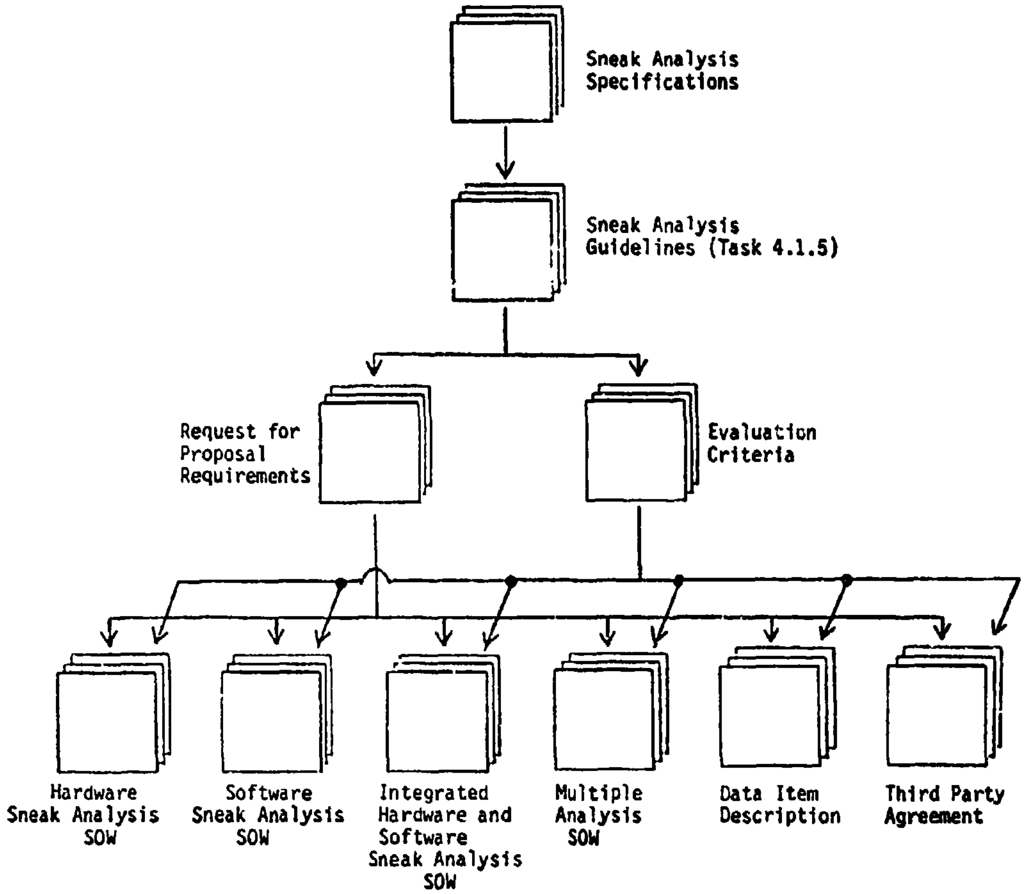


Figure 3-6. Selection Process for Sneak Analysis

3.4.1 Specification requirement for Snea[illegible]ysis. This specification is written for a combined hardware/software task an[illegible] may be edited to suit either task individually. In addition, the specification is written for an expanded project phase usage, even though the Appendix A Sneak Analysis Project History Tables contain past projects performed from the Full-Scale Engineering Development phase through the Unlimited Production phase. (A complete specification is provided in Appendix I.)

3.4.1.1 Purpose: The Sneak Analysis technique described herein establishes a standard procedure for performing the analysis and reporting the results. The analysis is used to systematically identify and report sneak paths, sneak timing, sneak labels, and sneak indicators which may exist in the design. All areas of design concern and document errors discovered during the Sneak Analysis are also reported. Such sneak conditions and design concerns that are discovered are to be assessed for their impact on system performance.

3.4.1.2 Usage: Sneak Analysis is a powerful tool to identify system conditions that could degrade or adversely impact the mission safety or basic equipment reliability. It is a technique that can be applied to both hardware and software. For hardware, SCA is generally based on the use of engineering and manufacturing level documentation. Its purpose is to identify latent paths which cause occurrence of unwanted functions or inhibit desired functions, assuming all components are functioning properly. SCA of electrical circuits is a mature and useful technique that can be performed on both analog and digital circuitry. The purpose for Software Sneak Analysis is to define logic control paths which cause unwanted operations to occur or which bypass desired operations without regard to failures of the hardware system to respond as programmed. After a Sneak Circuit Analysis and a Software Sneak Analysis have been performed on a system, the interactions of the hardware with the system software can readily be determined. The effect of a control operation that is initiated by some hardware element can be traced through the hardware until it enters the system software. The logic flow can then be traced through the software to determine its ultimate impact on the system. Similarly, the logic sequence of a software initiated action can be followed through the software and electrical circuits until its eventual total system impact can be assessed. Finally, the analysis should be considered for critical systems and functions where other techniques are not effective, but should not be applied to off-the-shelf computer hardware such as memory or data processing equipment.

Sneak Analysis is a useful engineering tool which, for hardware, can be used to identify sneak circuits, drawing errors and design concerns. Software analysis will identify software sneaks, design concerns and software document errors. The effects of varying environments are not normally considered, and sneak circuits which result from hardware failure, malfunction, or environmentally sensitive characteristics are not usually identified. The identification of a sneak circuit does not always indicate an undesirable condition; in fact, some have been used to accomplish tasks when other circuitry has failed. The implications of a sneak circuit, therefore, must be explored and its impact on the circuit function determined before any corrective action is taken.

3.4.1.3 Applicable documents: The following documents of the issue noted or, if not noted, the issue in effect as of the date of the contract as shown in Department of Defense Index of Specifications and Standards form a part of this specification to the extent specified in Table 3-34.

TABLE 3-34. APPLICABLE DOCUMENTATION

DOCUMENT NUMBER	DOCUMENT NAME
1. MIL-STD-785B (15 September 1980)	Reliability Program Plan for Systems and Equipment
2. MIL-STD-781C (21 October 1977)	Reliability Design Qualification and Acceptance Test Standard (Appendix A, Para. 40.7)
3. MIL-STD-882A (28 June 1977)	System Safety Program Requirements (Para., 5.5.1.2.c)
4. NAVSEA TE001-AA-GYD-010/SCA	Contracting and Management Guide for Sneak Circuit Analysis (SCA)
5. DOD 5000.40 (8 July 1980)	Reliability and Maintainability (Para. D.2a)

3.4.1.4 Requirements: During the Full-Scale Engineering Development and/or Unlimited Production phase, the contractor shall conduct or contract for Sneak Analysis of circuitry/software critical to mission success or crew safety. The analyses shall identify latent paths which cause unwanted functions to occur or which inhibit desired functions. Potential design or equipment weaknesses are to be identified and reported. An assessment of the system impact is to be provided for the potential problem, along with a recommendation for corrective action. In making these analyses, all components/software shall be assumed to be functioning properly. These analyses shall be performed on production manufacturing documentation for each circuit analyzed, and the actual code and specifications for the software to be analyzed during the Full-Scale Engineering Development or later phases. Equivalent or design type drawings or logic flows shall be used during earlier development phases.

The contractor shall present in the proposal a complete list or description of the functions/circuitry/software for which Sneak Analysis is to be conducted. The list of those functions/circuits/software to be analyzed shall be presented to the procuring activity, together with the rationale for any deviations to the specified systems.

The contractor shall specify the overall task period of performance along with subtask periods of performance. Periodic reviews or report periods shall be established to promote timely transmission and consideration of contractor reports. A final report documenting the task and all findings shall be prepared and transmitted at the conclusion of the task. If the Sneak Analysis task includes change analysis, a final report for the baseline analysis shall be required, followed at the conclusion of the change analysis task with a final report documenting the change analysis reports.

The contractor shall indicate the depth of the Sneak Analysis task in the proposal. The typical hardware or software task occurs at the component level or instruction level where cause and effect relationships are studied in detail. Systems defined as within scope of the contracted effort should be analyzed to the detail component level. An interface analysis should be performed for that portion of out-of-scope equipment or software directly interfacing with the in-scope systems.

3.4.1.5 Technique: Four classical categories of the Sneak Analysis technique shall be addressed:

a. Sneak paths, which allow electrical current or software logic to flow along unsuspected routes

b. Sneak timing, which causes functions to be inhibited or to occur unexpectedly as a result of timing or function sequencing

c. Sneak labels, which cause an operator to initiate incorrect stimuli

d. Sneak indicators, which produce ambiguous or false displays.

A formal Sneak Analysis shall involve (1) classifying basic circuit or software recognition patterns into which the system elements fall, (2) application of "clues," or sneak condition criteria, applied to these patterns to uncover sneak conditions, (3) assessing the effect of the sneak conditions on system performance, (4) establishing accept/reject criteria, and (5) reporting of results to the procuring activity.

The contractor shall identify latent flow paths, unexpected operational modes, unnecessary components, etc., in case of hardware; or unused and inaccessible paths, improper branch sequencing, undesirable loops, etc., in the case of software. The system Sneak Analysis should not be finalized until complete production system drawings are available. Any proposed changes made after production drawing release shall be examined for introduction of sneak effects before being adopted.

The contractor shall document the criteria, assumptions, delineation of sneak paths, etc., of the analysis. The data shall be maintained and updated to reflect any changes in equipment configuration.

3.4.1.6 Assumptions: Assumptions are made when performing Sneak Analysis to establish the analysis boundaries, define terminology, and keep the scope within cost-effective bounds. Tables 3-35 and 3-36 list the more common assumptions for hardware and software, respectively.

TABLE 3-35. HARDWARE SNEAK ANALYSIS ASSUMPTIONS

a. A Sneak Circuit is not dependent on a component or circuit failure.

b. Unless otherwise specified, signals which cross analysis boundaries (out of scope) are assumed to be correct in voltage, polarity, and time for the circuit being analyzed.

c. The data base for the analysis represents the "as-built" configuration of the system.

d. Parametric calculations are performed only to the extent necessary to understand true circuit operations.

e. Environmental effects are not normally considered in the analysis.

TABLE 3-36. SOFTWARE SNEAK ANALYSIS ASSUMPTIONS

a. The software specification is the design intent of the software.

b. The assembler or compiler does not introduce errors into the software.

c. Assembled or compiled software is free of syntax errors, i.e., typographical errors.

d. The data provided represents the complete software program under consideration.

e. Hardware induced problems are not considered.

3.4.1.7 Quality assurance provisions: To perform a valid Sneak Analysis, provisions must be established to provide that: (1) all paths within a system have been analyzed; (2) each path is directly traceable to the network tree in which it was analyzed; (3) each component/statement is directly traceable to the path in which it was analyzed; and (4) each component/statement is directly traceable to the specific documentation used to establish the data base master file.

The following provisions shall be used to produce a valid Sneak Analysis:

1. Network trees analyzed for sneak conditions shall be traceable to the system's manufacturing drawings or source code. The network trees shall contain all the wiring, components, or statements used to generate the tree. Further, all paths necessary to initiate and complete a given function shall be shown or referenced on one network tree.

2. Each network tree shall be independently numbered.

3. An index shall be developed to show the network tree in which each component or statement appears.

4. Each path shall be traceable to the network tree in which it appears.

5. Each path shall be independently numbered, and its wire segments, components, or statements traceable to the system's manufacturing drawings or source code in which they appear.

6. Each path shall be independently analyzed as to its effects on system operation, and records maintained indicating analysis results.

3.4.2 Sneak Analysis Request for Proposal considerations. The basic requirements for a procuring activity initiated Sneak Analysis Request for Proposal (RFP) are presented in this section. The outline can be tailored by the procuring activity to suit a specific application. The outline is intended to be specific enough for the procuring activity to properly assess and evaluate contractor responses. Evaluation criteria are also presented to provide a basis for evaluation.

The outline of the Sneak Analysis RFP and evaluation criteria are shown in Table 3-37.

TABLE 3-37. OUTLINE OF A SNEAK ANALYSIS REQUEST FOR PROPOSAL

REQUEST FOR PROPOSAL (RFP) OUTLINE	
1. PROGRAM NAME 2. PURPOSE OF RFP 3. SCOPE OF EFFORT 4. APPLICABLE SUBSYSTEMS 5. ANALYSIS DEPTH 6. CHANGE ANALYSIS OPTION 7. ADDITIONAL ANALYSES	8. DATA REQUIREMENTS 9. TASK DESCRIPTIONS 10. DELIVERABLES 11. MISCELLANEOUS 12. FACILITIES AND SECURITY 13. COST 14. TIME REQUIREMENTS
EVALUATION CRITERIA (EC)	
1. UNDERSTANDING PROBLEM 2. RELEVANT PAST PERFORMANCE 3. CAPABILITY TO PERFORM	4. COST 5. SCHEDULE

3.4.2.1 Request for Proposal items:

1. Program Name - This is a title of the Sneak Analysis task which is generally the program name. The major subsystem equipment or software name may be added to the title.

2. Purpose of RFP - The purpose(s) for requesting the Sneak Analysis should be stated. The rationale may be oriented toward problem identification or design analysis at the Validation and Full-Scale Engineering Development phases, or problem identification or change analysis in later development phases. This section should stipulate the task as being a "one-shot" analysis, a continuing analysis with change analysis included, or a combination of Sneak Analysis with one or more analysis techniques. Combined hardware and software Sneak Analyses are to be stipulated as an Integrated Analysis.

3. Scope of Effort - The task scope should delineate the task requirements, depth of analysis, the system or subsystems included in the analysis, the period of performance, and the end product deliverables. The scoping paragraphs may contain important notes or clauses from the remaining RFP items described in this section. Specific systems or subsystems may be excluded from the effort and listed in this section. If multiple systems or subsystems are to be analyzed one at a time, the order and time phasing for each subtask should be specified. The responsibility

for data acquisition should be identified as a task for the procuring activity, the Sneak Analysis contractor, or a third party. When classified systems are to be analyzed, security requirements should be specified for personnel, facilities, documentation handling procedures, and computer processing. The task scope should specify whether a change analysis is included in the overall effort, and, if so, the number and type of acceptable changes should be specified.

4. Applicable Subsystems - Applicable subsystems are to be accurately and completely identified in the RFP. Figures and pictures may be used to clarify and bound the applicable areas. Accuracy is required in this item because cost, schedule or problem identification limitations may require analysis of only portions of particular subsystems and whole subsystems in others. Whenever portions of primary functions in the applicable subsystems continue into equipment or software not considered in scope, it is necessary to provide interface control documents, functional system diagrams, or logic type diagrams that identify or depict the remainder of the function.

5. Analysis Depth - Analysis depth is an important scoping consideration because it has a direct bearing on cost, schedule, and anticipated results. The procuring activity should specify the level of the Sneak Analysis required as interface or component level. Interface analysis concentrates on high level system functions, while component analysis proceeds into the detailed subsystem equipment. This RFP item is important to the procuring activity because it enables a correlation of RFP responses. Some responses may cover the designated subsystems and other RFP requirements for markedly lower cost and schedule time due to performance of a higher level systems analysis rather than a more detailed systems analysis. The analysis depth specified in the RFP must be matched to the level of detail in the acquired documentation.

6. Change Analysis Option - The incorporation and analysis of electrical system equipment changes or software code instruction changes must be specified in the RFP if desired by the procuring activity. The change option may be limited to an analysis for only the proposed wiring or software design changes brought about by prime contractor response to Sneak Analysis reports. A more formal change analysis would include all changes to a particular configuration baseline. In either case, the procuring activity should specify the number and type of changes desired and the change analysis period of performance.

A majority of change analysis projects occurred in the Space Environment grouping of the Appendix A Sneak Analysis Project History Tables. These were relatively long duration projects in comparison to the Airborne and Ground/Water projects. Typical project phasing for a formal change analysis is during the Full-Scale Engineering Development Phase and Full-Scale Prototype Development Phase. The rate and number of changes are typically high for these phases. By the Pilot Production and Unlimited Production Phases, the project design has been firmed and under configuration management control. The rate of change generation is typically lower in these two phases.

7. Additional Analyses - Whenever Sneak Analysis is to be performed in conjunction with other analyses, the selection and phasing of the tasks should be specified. Care should be exercised in this process to ensure that the maximum benefits are achieved for each of the analyses. Combined analyses are usually performed on areas of circuitry or software that are new in concept or design. They may also be applied to areas which have been manifesting anomalies. Combining analyses is also a means of achieving cost reductions. Central to any analysis is the effort required to establish the configuration on which the analysis is to be performed. A combined analysis effort requires the design configuration building only once, and this has typically been accomplished by the Sneak Analysis task for electrical and software systems. The Sneak Analysis network tree approach provides a functional layout of the system where cause and effect relationships can be identified and depicted easily. While the fundamental premise of Sneak Analysis assumes no equipment or software failures, failures can be postulated at any level and their effects traced through the systems in the same way as a detailed Failure Mode and Effects Analysis. The network trees also serve as the basis for other analyses, which eliminates duplicate efforts, standardizes the configuration, and lowers the overall costs.

8. Data Requirements - The data requirements for Sneak Analysis depend on the task type and analysis depth. An interface analysis requires interface control documents, system block diagrams, high level functional diagrams and program description documents, including requirements and specifications. A component analysis requires the logic flow diagrams and functional integrated schematics or flowcharts, if they exist, along with the documentation required for an interface analysis. The analysis also requires source code listings, detailed schematics, cable interconnect diagrams, wire lists, printed circuit network drawings, and in some cases, assembly drawings, procedures and check-lists.

The procuring activity must assign the data acquisition task responsibility to the prime contractor and subs, to the procuring activity itself, or to the performing Sneak Analysis contractor. The procuring activity, working through the prime contractor and vendors, has been the customary designee for this task. Allocations for the data acquisition effort will be made regardless of who is selected. Items to be considered for this effort are the prime contractor and vendor costs and delivery schedules if they are designated for data acquisition. The procuring activity and the Sneak Analysis contractor require funding to identify, order, and possibly travel to acquire the documentation.

The procuring activity should verify that the overall program contract has requirements for prime contractor, subcontractor and vendor deliveries of drawings and software code which will permit analyses (like Sneak Analysis) to be performed in a timely and cost-effective manner. If delivery of drawings and code has not been contracted for in the overall program procurement process, additional cost will be incurred by the procuring activity in obtaining drawings and code for the analysis. Third party agreements will also have to be established which identify the documentation users, the purpose, data handling procedures, period of usage and final disposition of the documentation.

Documentation acquisition must be timely and complete early in the Sneak Analysis task or schedule impacts will occur. From the results of Task 2, the majority of Sneak Analysis tasks are under nine months in duration which means that the complete documentation for such a task should be received within a month of task initiation. Definite milestone dates indicating 50% and 100% levels for data acquisition should be stated in the RFP.

The RFP should specify when the analysis is to include classified systems so that special handling procedures are instituted. Personnel availability with proper security clearance levels must be established, along with facility clearances. If data processing is to be performed using classified data, then the computer and computer facility must have been approved for classified data processing.

9. Task Descriptions - The RFP should stipulate that the competing Sneak Analysis contractors provide a description of the tasks they are to perform to identify sneak conditions in hardware and/or software. The approach developed on the Apollo project and refined through numerous Sneak Analysis projects is to use network trees in combination with clue check-lists to identify problems. The approach should be systematic, thorough, and complete. The approach should demonstrate to the procuring activity that each and every circuit function will be scrutinized in detail. The intent should demonstrate that intended functions

are turned on and off at the proper time and for the proper duration, and also that unintended functions are not generated. It is in the unintended function generation or inhibition that Sneak Analysis excels. Sneak Analysis pursues functions from the component level operations up to system level effects (as in Failure Mode and Effects Analysis) and from top level system components down to triggering components (as in a Fault Tree Analysis). Sneak Analysis also looks across component levels for additional system effects.

The task description should also declare whether any automation aids are contemplated for use in accomplishing the Sneak Analysis task. Descriptions of the automation aids should be provided including the rationale for use. Relevant points to consider for automation include:

a. Direct conversion of prime contractor manufacturing data from magnetic tape (or other suitable media) to acceptable input format for the Sneak Analysis computer programs.

b. Generation of the network trees used in the analysis phase.

c. Timing analyses.

d. Some circuit analysis.

e. Change analysis.

10. Deliverables - The deliverables of a Sneak Analysis task should be specified in the contractor proposal, along with concise descriptions of each item. Descriptions of these reports are provided in the example Data Item Descriptions provided later in this section. As a minimum, the following output reports should be included:

a. Periodic Status Report - This report documents the progress of the various Sneak Analysis subtasks, identifies any problems encountered in the analysis which might impede successful completion of the project, tabulates all Sneak Reports issued, and includes copies of the Sneak Reports as they are issued. The Status Report is the primary means of conveying the findings of the task in a timely fashion. The majority of projects included in the History Tables required either bi-weekly or monthly status reports. Long duration projects, especially in the Space Environment, also called for special quarterly reports which summarized the activities of the previous three-month period.

b. Sneak Analysis Reports - This category of reports includes Sneak Circuit Reports, Design Concern Reports, and Drawing Error Reports for hardware type systems, and Software Sneak Reports, Software Design Concern Reports, and Software Drawing Error Reports for software type systems. An example of each type of report is presented in Appendix J. The primary intent of each report is to document a potentially serious system problem in a form that is understandable, clear, concise, and easily verifiable. The paperwork should serve as a link from the Sneak Analysis contractor to the procuring activity and on to the prime contractor. The Sneak Analysis Report should briefly describe the undesirable circuit or software condition, the potential crew or mission impact, all relevant documentation so that the basic system configuration can be verified, a figure depicting the actual condition, and a proposed recommendation to eliminate the system problem.

c. Final Report - This report summarizes the entire activities of the Sneak Analysis task. The report should include the project purpose, scope, and an overall assessment or evaluation of the analyzed system(s). A complete listing of all documentation used in the analysis should be provided so that the configurations for the baseline system and system changes can be established. A brief description of the project tasks might be desirable, even though they may be referenced back to the originating project RFP. A tabulation of all reports issued by the Sneak Analysis contractor to the procuring activity should also be included, along with copies of each report. Any problems which had an impact on the successful completion of the task according to the scope and terms of the originating RFP should also be documented in the Final Report.

The three report types just presented are the minimum deliverables for a Sneak Analysis task. If Change Analysis is to be performed, any problem conditions can be reported sufficiently with the above type reports. If a verification of check lists, T.O.'s, or other military type procedure lists are included within the scope of the contracted effort, slightly modified versions of the Sneak Analysis Reports may be desirable. In this instance, the reports would identify operator induced errors, errors in control sequencing, and errors of interpretation and reaction to equipment displays. Additional reports generated to document the results of other analysis techniques should also be described and included in the Final Report.

The Final Report could also be used to implement the recommendation to measure Sneak Analysis effectiveness, which is included in Section 5.0 of this report.

11. Miscellaneous - Miscellaneous items which are required of the Sneak Analysis contractor over and above the baseline analysis should be specified. This may include, but not be limited to, the following:

    a. Additional analyses, such as Failure Mode and Effects Analysis, Fault Tree Analysis, and Change Analysis.

    b. Data acquisition and possibly some travel.

    c. Program review support to provide liaison and inputs for milestone events such as a PDR, CDR, first flight, or scheduled tests.

    d. Computer program development, possibly in the area of converting prime contractor data to a format usable in the analysis phase.

    e. Classified data handling.

12. Facilities and Security - Facilities and security requirements should be a part of the RFP, even when classified data systems are not involved. This feature protects the proprietary rights, if any, of the manufacturing contractor. The entire contractor facility or some designated portion thereof should be cleared to handle the highest level documentation contemplated for use by the Sneak Analysis contractor. Personnel should have at least corresponding security clearances. This includes direct management, engineers and software analysts, clerks, secretaries, aides, and any other personnel participating in the classified portion of the analysis. If any portion of the task is computer-aided, then the personnel, computer and computer facility must also be cleared. Dispositioning requirements to return or destroy acquired documentation and Sneak Analysis contractor generated reports must be specified.

13. Cost - Cost breakouts for the various Sneak Analysis tasks will normally be a separate report or volume from the RFP to allow an impartial evaluation of the technical portion of the RFP. Cost factors include analysis personnel, computer processing, classified data handling, data acquisition, travel, performance of other analyses, and computer program development (if any).

14. Time Requirements - The time requirements (if any) of the procuring activity should be stated in the RFP. Dates for support of CDR, first flight or major systems tests have a direct bearing on the tasks and sequence of tasks performed by the Sneak Analysis

contractor. If the analysis is "one-shot" and scoped to a single system or portion of a single system, then the task approach is fairly direct. Otherwise, multiple systems analysis or combined analyses will require the contractor to prioritize the systems and the tasks within each system.

The RFP should also require a schedule of all major functions to be performed by the Sneak Analysis contractor. The schedule should show an orderly progression of tasks leading to successful completion of the analysis, and any intervening milestones specifically required in the RFP.

3.4.2.2 Evaluation criteria items:

1. Understanding Problem - Understanding of the problem is an important evaluation criterion to judge Sneak Analysis proposals. The evaluation criterion looks not only for an understanding of the contractor's Sneak Analysis process and task relationships but also the need and use of the analysis to satisfy the procuring activity's requirements in a timely and cost-effective manner.

The task process may be different from contractor to contractor, but will probably progress according to the following order:

a. Data Acquisition

   1. Identify and acquire data if so tasked.
   2. Log all documentation received.
   3. Review all documentation for completeness and correct level of detail.
   4. Identify missing and required documentation.

b. Preanalysis

   1. Identify functions in the documentation.
   2. Verify overall system interconnections.
   3. Exclude all areas of documentation out of scope for the effort.
   4. Review adequacy of interface equipment or software documentation.

c. Partitioning

   1. Subdivide circuitry or software by functions.
   2. Annotate documentation for subsequent encoding task.

d. Encoding (only if computer processing is used)

   1. Convert detailed wire continuity segments to computer format.
   2. Convert detailed source code to computer format.

3. Automate conversion process
4. Verify data masterfiles reflect actual system configuration.

e. Computer Processing

1. Edit and analyze all user inputs.
2. Connect all circuit segments and software code.
3. Produce hardcopy plots of system network trees.

f. Network Tree Generation

1. Manually develop network trees if no automation aids are used.
2. Annotate all functional remarks on network trees.
3. Annotate all cross-references on network trees.
4. Identify and annotate relevant descriptive documentation.
5. Annotate interface information for out-of-scope systems.

g. Analysis

1. Identify topographs (circuit or software patterns) in network trees.
2. Apply "clues" for each topograph.
3. Compare network trees to functional system flows.
4. Compare network trees to Interface Control Documents.
5. Compare network trees to procedure check lists.
6. Perform timing analyses where required.

h. Problem Reporting

1. Assess problem categorization (Documentation, Design, Sneak).
2. Identify relevant documentation.
3. Describe problem.
4. Determine system impact of problem.
5. Provide sketch of system illustrating problem.

i. Status Reporting

1. Provide periodic status reports (bi-weekly or monthly).
2. Provide quarterly status reports, if required.
3. Include all reports issued during the period.
4. Tabulate and identify report dispositions.

j. Change Data Receipt (if change analysis option is included)

1. Acquire proposed and/or implemented system changes.
2. Log all documentation received.
3. Group changes by function or configuration control board package numbers.

k. Change Data Incorporation

1. Determine extent of change package.
2. Assess automation requirements if change is significant.
3. Update network trees.

l. Change Data Analysis

1. Re-analyze all changed network trees.
2. Re-analyze all network trees affected by the changed network trees.
3. Rerun timing analyses, if necessary.
4. Problem reporting is the same as baseline analysis.

m. Final Report

1. Summarize task purpose and scope
2. Describe analysis technique.
3. Provide composite listing of all documentation.
4. Provide tabulation of all reports and dispositions.
5. Provide copies of all reports.
6. Provide task assessment.
7. Provide additional task or system recommendations.

2. Relevant Past Performance - Relevant past performance is considered to be a very important evaluation criterion. Actual performance, along with an assessment of that performance, should be demonstrated and presented in the contractor's proposal. The distinction between actual performance and performance capability needs to be identified and evaluated in the contractor's response. To adequately evaluate responses, the contractor should be required to provide task synopses similar to those required by the Application Guidelines procurement and in addition, provide upon request, copies of Final Reports for procuring activity inspection.

While the overwhelming majority of Sneak Analysis projects were performed successfully, some projects were adversely affected by incomplete and late data acquisition, technique development and personnel learning curves. Sneak Analysis, as with other analyses, requires a large expenditure in personnel development and training, technique development, task phasing, computer program development,

and new technology incorporation. Relevant past performance is thus essential for successful accomplishment of the Sneak Analysis task.

3. Capability to Perform - Capability to perform Sneak Analysis is the evaluation criterion which will be the most difficult to assess. Different approaches will be offered in competitive proposals by prospective contractors. Some may include the formal Sneak Analysis process, variations to the process, or substitution of other analysis techniques to achieve the same end result. As a minimum, the capability to perform the basic tasks listed in Item 1 would be a prerequisite for further proposal evaluation. Some of the main task headings may change, but the overall tasks should be equivalent. The next major evaluation point will center on the analysis technique itself. Since the major function of the analysis is problem identification, many diverse approaches can be offered, and justifiably so. Problems can be found by virtually any analysis technique, but the overriding evaluation considerations are consistency, systemization, and a unique perspective and perception of circuit and software functions. Current automation aids and descriptions of their intended use should certainly strengthen the evaluation rating in the capability to perform criteria.

4. Cost - An evaluation of cost factors is especially important in Sneak Analysis tasks, because cost has an important bearing on the scope and depth of the analysis and vice-versa. Line item costs for the basic analysis, change analysis and other analyses should be separately reported in the proposal. Costs for personnel, management and clerical support should be delineated, as well as, computer processing costs, travel costs, and any other miscellaneous cost items. In general, Sneak Analysis has been sold on the basis of the number of components in the electrical system and the number of executable instructions in a software program. The greater the number of components and instructions, the greater the Sneak Analysis cost. However, cost needs to be evaluated against proposed depth of analysis. An interface analysis cannot be expected to cost as much as a detailed system analysis, for example.

5. Schedule - Contractor schedules should be evaluated for the proper sequencing and duration of tasks. They should identify all major task functions and project milestones. Since data acquisition is very critical, early milestones for data receipt are an absolute necessity. Network tree construction can either be shown as one complete task of relatively short duration in the first half of the project, or as a continuing process throughout most of the period of performance. Either approach is acceptable. Analysis is the main task element of the procurement. The longest duration task should be the analysis, but the schedule duration may not indicate such emphasis. Contractor discretion

to man-load the task at this phase is certainly acceptable and desirable. In addition, the functions listed on the schedule are normally not that discrete in the performance of a Sneak Analysis task. That is to say, some analysis is normally performed in each task shown on the schedule.

The contractor schedule should also include reviews (both program design and status), other analysis technique sequencing and duration, change analysis if selected, and Final Report preparation and delivery. If classified data and proprietary data are used, there should be a final data dispositioning task included.

3.4.2.3 Contract selection: Selection of the type of contract desired for the Sneak Analysis procurement is primarily the option of the procuring activity. However, the experience base acquired from the Appendix A Project History Tables indicates that the predominant contract type is Firm Fixed Price (FFP), as shown in Table 3-38.

TABLE 3-38. SNEAK ANALYSIS CONTRACT SUMMARY

CONTRACT TYPE / PROJECT ENVIRONMENT	COST PLUS FIXED FEE		FIRM FIXED PRICE		TOTAL
	NUMBER OF PROJECTS	%	NUMBER OF PROJECTS	%	NUMBER OF PROJECTS
SPACE	8	45%	10	55%	18
AIRBORNE	6	10%	55	90%	61
GROUND/WATER	3	10%	27	90%	30
COMPOSITE	17	16%	92	84%	109

Table 3-38 illustrates that long duration projects (approximately one year or longer) and projects with significant change analysis options are more likely to be released as Cost Plus Fixed Fee (CPFF) contracts. In addition, those projects which require contractor risk to accomplish unusual tasks are also likely to result in CPFF contracts. The Space Environment is the primary project environment for implementing CPFF contracts, with approximately half of the project contracts so designated.

Short to medium duration projects (less than one year) are predominantly issued as FFP contracts. The Airborne and Ground/Water environments result in equivalent distributions with 90% FFP contracts. FFP contracts limit the procuring activity's liabilities costwise in the performance of Sneak Analysis.

The type of contract should be stipulated in the RFP.

3.4.3 Tailoring Statement of Work requirements. The process for tailoring Sneak Analysis Statement of Work requirements to acquisitions of various types is shown in generalized form in Figure 3-7. The process assumes that an assessment of specification requirements and application guidelines (Task 5, Section 3.5) resulted in a decision to perform the Sneak Analysis task. The need and the rationale for the task are thus established. The process flow is now directed toward determining how the procurement is to be implemented.

3.4.3.1 Tailoring process: The process flow shown in Figure 3-7 will be presented as a step-by-step description of each function or decision block. The blocks are numbered to clarify the path direction within the decision tree.

1. Identify Analysis Areas - The process begins in Block 1 with an identification of relevant systems or subsystems. On the initial pass, entire systems may be scoped to determine overall Rough Order of Magnitude (ROM) costs. All equipment or software within each of the designated systems would be identified by name, including component equipment and interconnecting cables for hardware systems and main routines and called subroutines for software systems. A high level interconnect diagram and parts list are necessary for this step.

2. Estimate System Size - Block 2 requires an estimation of the number of instructions in the software system. Typical systems are composed of discrete devices and hybrid devices in hardware and high order languages, assembly languages and machine code in software. Each device or instruction can be translated into an equivalent number of discrete components or assembly language instructions, respectively.

   The overall number of components by category should be summed and used in the next process block.

   In the hardware estimation process, only equipment devices are counted, not the interconnecting cabling. In the software estimation process, only executable code is counted, comment code is not.

3. Compute Cost - Tables for computing costs for Sneak Analysis tasks are included in Appendix B. The listed costs are derived for a 1979 dollar basis. The cost tables are intended to show the cost per device type. By knowing the costs per device/instruction and the number of devices/instructions, the ROM costs can be determined.

   The cost tables have been evaluated and determined to be adequate for estimating a ROM cost for Sneak Analysis. The budgetary estimate is for planning purposes only. Experience has shown that the final price for the performance of Sneak Analysis is frequently lower when the actual drawings and/or source codes are used for price determination. In addition, the final project price may be lower when discrete functions are identified within the overall systems and the analysis is limited to those functions.

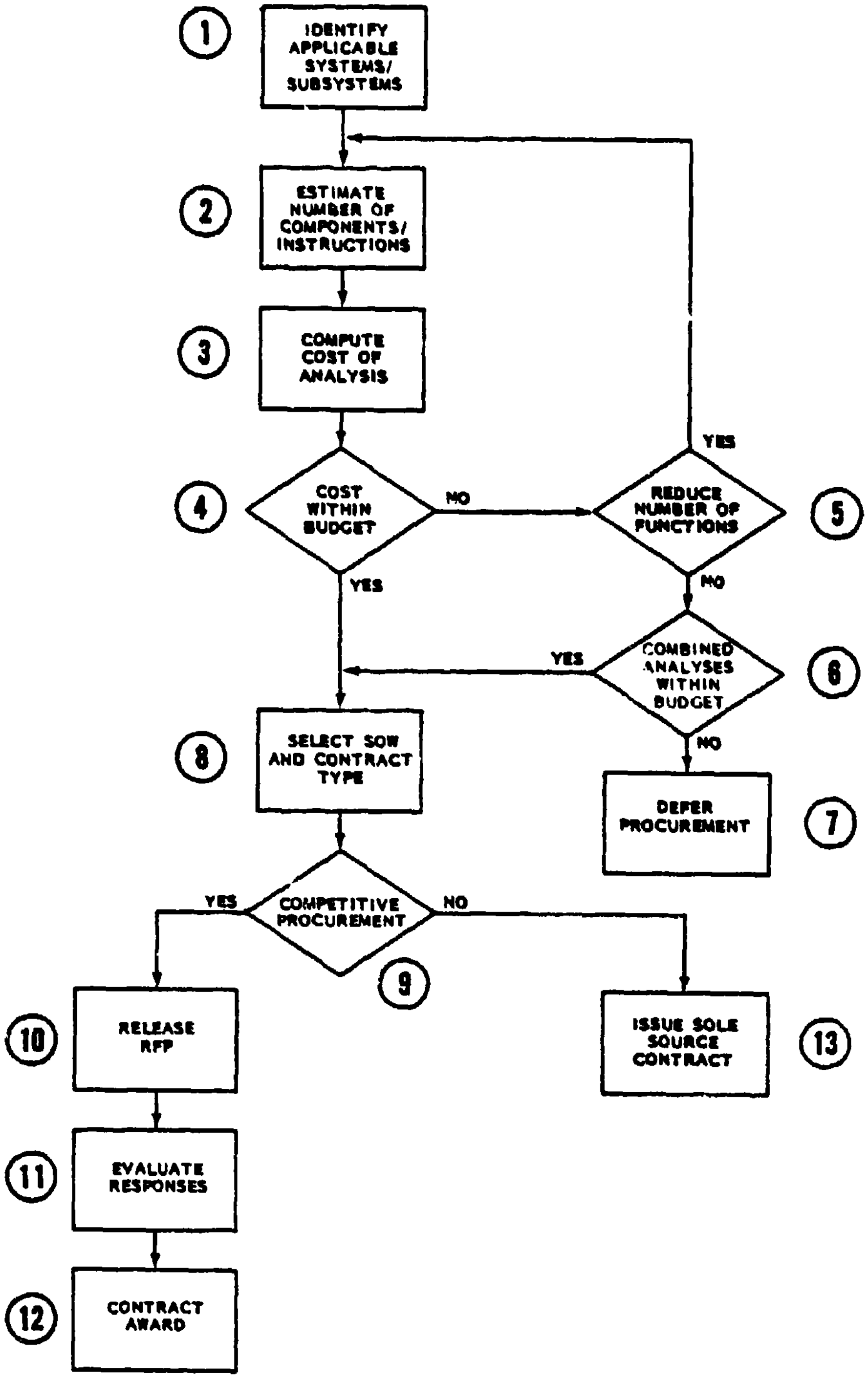


Figure 3-7 Tailoring Statement of Work Requirements

4. Acceptable Cost Range - The derived ROM cost of the Sneak Analysis task is compared to the budget levels allocated in the original program reliability plan. If anticipated costs are within the budgeted level, the procuring activity may begin the procurement process by proceeding to the function in Block 8 of Figure 3-7. However, since the original estimate was a ROM, it may be necessary to perform a more detailed estimate of costs to achieve higher confidence in the cost figures. It should be noted that the derived cost is actually the center point of a cost range. For example, if the derived cost is $50,000 ± 10%, the cost should be between $45,000 to $55,000.

   If the task cost significantly exceeds the allocated analysis cost, additional decisions must be made by considering reduction of analysis scope or combining Sneak Analysis with other analyses.

5. System Scope Reduction - If the ROM estimate exceeded the planned task allocation, a critical analysis of the designated systems should be performed to isolate the desired functions. In effect, the task effort should be scoped from the overall system level to the subsystem level and possibly even to the component level. Non-critical functions should be eliminated and a new scope of effort determined. The reduced task scope may then be evaluated against the original statement of need and costs recomputed by entry to Block 2. Process cycling through scope reductions and cost computations may occur until desired task cost levels are achieved.

   Note - The procuring activity could request contingency funds to supplement the original task allocations. Program contingency funds are those dollars not previously allocated to other analyses. Sufficient budget could then be obtained and the process flow could proceed to Block 8.

6. Combine Analyses - If it is not possible to reduce the number of functions scoped for the analysis task, the procuring activity should consider combining Sneak Analysis with other analysis tasks. The overall cost of performing Sneak Analysis with specific analyses such as Failure Mode and Effects Analysis and Fault Tree Analysis, is lower than performing all three tasks separately. In this way, lower overall budget expenditures can be obtained for the combined tasks. The resulting allocations should be sufficient for the combined tasks. If the resulting budget allocation is sufficient, proceed to Block 8, otherwise the Sneak Analysis task must be deferred until supplemental funds are available or the procurement cancelled. An option is available to the procuring activity that removes the requirement for a detailed Sneak Analysis and imposes the requirement for an interface Sneak Analysis. Very early in the program development phase (Concept or Validation Phase) this may be a viable alternative. From the Full-Scale Development Phase and later phases, the analysis should be at the detailed level to obtain significant results.

7. Defer Procurement - If this task decision block is reached, a reconsideration of rationale or statement of need should be made. If the decision is made to defer the task, periodic reviews should be performed to re-institute the proposed task at the earliest available time. The cost to identify problems, correct design and implement changes increases dramatically when problems are detected in later project development phases.

8. Contract Selection - After the Sneak Analysis task is properly scoped and program funds are allocated, the next function to be performed is selection of the type of contract and statement of work. Table 3-38 presents types of contracts used in past analyses and may be used as a basis for the pending procurement. The selection of the proper statement of work is dependent on the equipment or software composition and the addition of other task analyses. Reference Section 3.4.3.2.

9. Competitive Procurement - The decision to issue a sole source procurement or a competitive bid procurement can probably be made at an earlier time in the process flow. If sole source selection is desired, proceed to Block 13 of the process flow. If a competitive bid procurement is selected, the decision should be made earlier in the development flow to overcome the attendant delay in the project initiation schedule until contract award.

10. Issue RFP - RFP considerations are presented in Section 3.4.2.1 which are relevant to the RFP structure. The bid package should then be circulated or announced in publications such as Commerce Business Daily.

11. Evalaute Responses - Evaluation criteria considerations are presented in Section 3.4.2.2 and may be used as the basis for evaluating contractor responses.

12. Contract Award - Contract award to the winning bidder is the final step in the competitive procurement process and represents the initiation of the Sneak Analysis task.

13. Sole Source Award - Sole source contracting for Sneak Analysis is the most common procurement method based on the Project History Table entries. This method offers a more immediate task startup and lower overall program incurred costs since the competitive procurement time phasing and expenses are eliminated.

3.4.3.2 Statements of Work (SOW): SOW descriptions are presented in this section, along with a brief discussion of the SOW contents. Example SOW's are provided in the following Appendices C, D, and E. The types of SOW's presented include:

1. Hardware Sneak Circuit Analysis - example provided in Appendix C.

2. Software Sneak Analysis - example provided in Appendix D.

3. Integrated Hardware/Software Sneak Analysis - example provided in Appendix E.

4. Combined Analyses - this is a variation of Appendices C, D, or E.

The typical content of a Sneak Analysis SOW (hardware and software) includes the following items:

1. General Purpose
2. Scope
3. Optional Change Analysis
4. Task Descriptions
5. Output Reports
6. Input Data Requirements
7. Period of Performance

   1. General Purpose - This item includes the task title and overall analysis objectives. If hardware only is involved, the example SOW in Appendix C is applicable; if software only, Appendix D applies; and, if hardware and software, Appendix E applies. The boilerplate SOW's do not include the project phase designator which would be added if formal tracking of Sneak Analysis effectiveness is to be performed.

   2. Scope - The scoping section is one of the three primary SOW tailoring items. For depth of analysis, an implicit assumption is made that a detailed component or instruction analysis is to be performed. Certainly the limits or bounds of the system or software are to be expressed in sufficient detail so that no doubt exists as to the functions or systems defined as in-scope. This bounding produces the number of components and instructions. If change analysis is to be performed, an extra sentence should be added which dictates the analysis and references item 3, Optional Change Analysis. Additional scoping parameters include other analyses, classified data systems, and data acquisition responsibilities.

   3. Optional Change Analysis - The change analysis option should be viewed in two ways. The first way includes the no-cost incorporation of engineering changes to the hardware or software prior to a designated task date. This task date has typically been referenced to a time period one to two months before the baseline analysis computer processing phase. After this time, the baseline is firmed and incorporation of changes is no longer within the original cost structure, with one exception. Changes, proposed or implemented, resulting from a Sneak Analysis report, can be performed at no cost subject to the number and relative size of the changes. In the second way, change documentation received after the start date and before a designated end date will be incorporated and analyzed if this additional cost option is selected. The performing contractor has two approaches for change analysis, which includes

manually incorporating the changes on the network trees or incorporating the changes into the data masterfile, running the computer programs and generating revised network trees. Only the changed network trees would be analyzed. However, the volume of changes and/or the size of individual changes is a distinct cost, schedule and performance risk factor for the performing contractor. To lessen this risk, the cost of the change option is typically tied to a percentage level of the overall baseline component or instruction count. The percentage level is variable and is directly proportional to change cost. Change analysis tasks are appropriately included in Cost Plus Fixed Fee contracts.

4. Task Descriptions - Task descriptions normally address in abbreviated form the generalized set of tasks shown in Section 3.4.2.2. Variations may occur in the overall set depending on the task application and scope.

5. Output Reports - Descriptions of the formal project reports may be provided in this item and may be referenced to the SOW Data Item Description. Sneak Analysis Reports and the Status and Final Reports are the primary components of this SOW item. A contract Data Requirement List will define the number of reports required, the delivery dates, and the recipients of the documentation. The Sneak Analysis reports may be omitted if the primary purpose of the analysis is to identify and describe a test or mission problem and when time is of the essence.

6. Input Data Requirements - Input data requirements are normally specified in a generic sense. The intent is to specify as-built or as-coded documentation, but no industry-wide standards exist and this introduces some risk to the performing Sneak Analysis contractor. Documentation defined as detailed by one contractor, subcontractor or vendor may be functional in nature to another supplier. The actual level and names of this "intended data" vary enough that absolute specification is not possible.

   All documentation may be received in a timely manner, but the schedule could be impacted if the right data, level of data, and current configuration data is not included. In general, the task schedule shows data receipt milestones, which, if they are not met, can result in a schedule impact and possibly additional cost to the procuring activity.

   The media for transmitting the data should be included.

7. Period of Performance - The period of performance item describes the main tasks shown on the project schedule with corresponding dates. Special program milestone data corresponding to PDR, CDR or first flight support can be included in this item.

3.4.3.3 Attachments to Statements of Work: Three attachments to Sneak Analysis Statements of Work are described in this section. The Data Item Description is a required contract item, submitted as an attachment to the procuring activity RFP. An example copy is provided in Appendix F. The third party (Proprietary) data agreement is an optional input and is used only when a contractor, subcontractor or vendor refuses to release documentation without assurances that the information will be safeguarded and its use restricted to the Sneak Analysis task. An example copy of this agreement is provided in Appendix G. The Task Schedule is the remaining attachment, and like the DID is a required SOW attachment. An example project schedule is provided in Appendix H. The project schedule should include as a minimum those tasks listed in Section 3.4.2.2 and the required task durations.

No tailoring is required for these three attachments, except possibly for task durations in the project schedule. The details of each attachment can be obtained by reading the example attachments.

3.4.4 Guidelines for monitoring Sneak Analysis tasks. This section presents the results for the second subtask of Task 4. Guidelines developed in this section are intended to acquaint the procuring activity with the necessary roles and functions it should perform to ensure successful accomplishment of the Sneak Analysis task. The guidelines contain active and passive functions.

The term procuring activity in this description is used to refer to both the contract administrator and the technical monitor. The technical monitor will perform the majority of functions.

The primary functions of the procuring activity include the following:

1. Contract performance
2. Data acquisition and handling
3. Liaison
4. Report evaluation, coordination and disposition.

3.4.4.1 Contract performance: This activity encompasses many functions designed to verify contractor compliance with the terms and conditions of the contract. It also includes contract extensions, modifications, add-ons, and redirections. The procuring activity should establish an orderly task startup by initiating contact between the procuring activity, the Sneak Analysis contractor and the data suppliers. During the task performance phase, the procuring activity should maintain the established communication lines to promote necessary information exchange. At the conclusion of the task, the procuring activity should verify final report receipt and content, prepare necessary closeout paperwork and verify all contract terms and conditions are met.

During the course of the contracted task, the periodic status reports will become the formal means of communicating task progress, problems and findings. The contractor status reports should be reviewed to ensure satisfactory progress on scheduled tasks and costs, and that these tasks are completed in a timely manner as indicated on the project schedule. Task slides in the data acquisition phase especially, and slides in the computer processing and network tree generation phases should be assessed to determine the impact on successful completion of the project. The procuring activity should also assess the likelihood of a task slide resulting in increased project costs. The procuring activity should actively investigate the causes of the slides and take actions to bring the tasks back on schedule.

The procuring activity should also verify all contract deliverables specified in the CDRL are received on time and in correct quantities. The deliverables should also comply with the DID items in effect for this procurement. If classified data is used in the task project, the procuring activity should verify proper safeguards and procedures are being used in the handling and transmission of classified material.

3.4.4.2 Data acquisition and handling: The procuring activity should maintain close contact in the startup phase with the Sneak Analysis contractor and documentation suppliers so that all input data requirements are satisfied. The procuring activity in particular should address the problems of incomplete data, illegible data and incorrect data. Active support of the procuring activity is required to expedite solutions to these data problems because of the typically short duration associated with the data acquisition task. If proprietary data agreements are required, they must receive the procuring activity's highest priority due to the long flow time in obtaining concurrence on third party agreements.

At the project conclusion, the procuring activity should verify that the proper disposition of the contractor supplied documentation is performed. Data may be destroyed according to approved procedures, returned to the data suppliers or the procuring activity, or maintained in the Sneak Analysis contractor files, depending on the conditions specified in the contract. All documentation sent to the Sneak Analysis contractor should be accounted for.

The standard operating procedures of the Sneak Analysis contractor should include a means of logging and recording all documentation received, including drawing numbers, revision levels, change notices, descriptive material, and program code. The contractor should supply a copy of this log to the procuring activity in every status report period in which data is received. In this way, the procuring activity is aware the data was transmitted and received, and can send this log to the document suppliers. The suppliers in turn should verify that the log is complete, correct, and reflects the current configuration for the proposed systems analysis. If it is not, the procuring activity should investigate the discrepancies and resolve the problem.

Sneak Analysis tasks involving a baseline analysis only require early data support by the procuring activity. Tasks which include change analysis require a continuing participation on the part of the procuring activity. It may be necessary for the procuring activity to be a formal member of the prime and support contractor change review board to ensure all change documentation is supplied to the Sneak Analysis contractor. In particular, the grouping and collecting of documentation changes associated with a single change directive is an important function of the procuring activity. Otherwise, receipt of drawing and software code changes associated with a particular change directive will be a disorganized and disruptive occurrence to the performing Sneak Analysis contractor. If possible, the procuring activity should see that the Sneak Analysis contractor is supplied with the formal change directive paperwork which specifies the revised hardware and software documentation. The Sneak Analysis contractor can then be assured that the documents received completely specify the system change and can be incorporated as a whole into the network trees and analyzed in an organized manner.

3.4.4.3 Liaison: The procuring activity should maintain a continuing line of communications with the Sneak Analysis contractor and the documentation suppliers. The role of the procuring activity will involve responding to questions by the Sneak Analysis contractor in regard to circuit and software operations, timing considerations, design philosophy, operating procedures and component specifications. Ideally, the procuring activity will possess a vast storehouse of knowledge and will answer all questions completely. In practice, however, the procuring activity becomes a focal point directing system questions to specialist personnel. The procuring activity should exchange communications between the two parties, but the text or results should be supplied verbally or in writing. The procuring activity will then be aware of the subject, the answers to the questions, and the disposition or status of any other open items. Misinformation or false assumptions can be detected early and corrected.

3.4.4.4 Report evaluation and disposition: This is the most important function of the procuring activity. The Sneak Analysis contractor has been contracted to identify and report systems problems and to transmit these results to the procuring activity. From here the procuring activity must establish a mechanism or procedure whereby the reports are critically assessed and evaluated and steps taken to create revised designs or procedures which correct the problems.

The process begins when the Sneak Analysis contractor notifies the procuring activity verbally or in the status report of potentially serious sneak conditions. Man critical or mission critical sneak conditions should be transmitted by telephone (assuming unclassified data) as the problems are identified and verified by the Sneak Analysis contractor. The written reports become an input in the next scheduled status report. The procuring activity should study the report in its entirety and should determine that the report is complete, legible, understandable and the problem areas highlighted. The title of the report should be descriptive of the sneak conditions, all relevant documentation required to configure the system through which the sneak passes should be listed, the ultimate system impact should be clearly stated, the problem description should be clear and concise, a design recommendation to eliminate the problem should be presented, and a graph or figure attached to illustrate the system elements and the problem.

When the procuring activity completes the report reviews, the reports should be disseminated to the program problem review board for consideration. Participation in the review cycle may be necessary to assure adequate review of the problems. The procuring activity should then maintain and track the reports through the review cycle to determine final disposition. When the reports are released to the manufacturers, the first task should involve a report review similar to that of the procuring activity. Next, the contractor should verify that the documentation numbers and revision status are current and correct, and that the drawing in the Sneak Analysis report is a correct rendition of the system. Then the report should be evaluated as to whether the condition is valid and possible. No assessment is required regarding the probability of the problem occurring. If the problem is deemed to be valid, the next task is to determine whether the postulated system effect is correct. Some testing and engineering judgment may be required to perform this task. The results of the entire investigation should then be documented and transmitted to the problem review board and a copy to the procuring activity. Recommendations for report disposition should be discussed, evaluated and documented, with the results sent to the procuring activity.

The process now may take different directions. If the report condition is valid and a recommendation is made to correct it, the report should be transmitted to the configuration control board for implementation. When design corrections have been approved and released by the board, the revised documentation along with the change authorization paperwork should be sent to the procuring activity for eventual transmittal to the Sneak Analysis contractor. The contractor in turn should examine the approved design changes by reanalyzing the revised configuration. The contractor should be looking to verify the sneak condition is eliminated, the design intent is accomplished, and no new sneaks produced. The evaluation results are then transmitted back to the procuring activity.

If the problem review board determined the report condition to be incorrect, the rationale should be documented and transmitted to the procuring activity and to the Sneak Analysis contractor. If incorrect documentation was the problem, the procuring activity should obtain the correct and current documentation and send it to the Sneak Analysis contractor. If the Sneak Analysis contractor is in error, the procuring activity should inform the contractor of the situation. The network tree construction process, the analysis technique training for systems analysts, or quality control procedures on the project tasks can then be improved and corrected. If a reanalysis by the Sneak Analysis contractor disputes the problem review board disposition, the procuring activity should consider a meeting of personnel to resolve the problem.

Report tracking should be instituted by the Sneak Analysis contractor. All reports to the procuring activity should be assigned a unique report number and recorded in a report log, along with the report title. This log should be a standard attachment which is updated for each status report. The report disposition should be designated as open or closed. Open status implies problem review board and change board consideration. Closed status implies a report disposition. The typically closed report categories are:

1. Hardware Change
2. Software Change
3. Procedure Change
4. Acceptable Risk
5. Cancel

Hardware and software changes affect the design documentation in all cases, and depending on the program development phase, actual hardware equipment and software code.

Procedure changes acknowledge the existence of the report problem, but an evaluation may determine that the problem can be adequately worked around (but not corrected). As long as the revised operating procedure is controlled and maintained, no problems should occur. But, realistically, the dormant problem can potentially recur due to design changes and design oversights. Procedure changes should not be encouraged as responses to any problem reports, especially in the early program development phases prior to unlimited production.

Acceptable risks also acknowledge the existence of the report problem, but either cost and schedule considerations prevail or an assessment as to a low probability of occurrence is given. In effect, nothing occurs other than the evaluation. The report is thus judged to be of no consequence.

The final disposition category is cancellation. This may indicate an incorrect Sneak Analysis assessment or incorrect documentation. The reason for cancellation should be determined and appropriate action taken.

3.4.4.5 Miscellaneous: The volume, type of reports, and report phasing are unique to each Sneak Analysis task. The Section 3.2 Task 2 results indicated an average number of reports by type per project. However, the actual distribution of reports experienced for a particular project may vary significantly from these averages due to many factors. A greater number of reports typically occur in the early program development phases and the volume rolls off in the Unlimited Production and Deployment phases. Report types (Sneak Reports, Design Concerns and Document Error Reports) also follow the same trend line as the number of reports.

One typical factor in the performance of a Sneak Analysis project is the timing of the reports. In the data acquisition and network tree generation tasks, the predominant type of report generated is the Drawing Error Report, which identifies discrepancies in the documentation and the contractor's assumption of the correct configuration. Design Concern Reports and Sneak Circuit Reports may be found at this time, but the numbers are low. Once the network trees are generated and the formal analysis task initiated, the predominant types of reports are Design Concern Reports and Sneak Reports. Few Drawing Error Reports are issued in this phase.

3.5 Task 5 - Sneak Analysis Application Guidelines. Task 5 - Develop an application guideline for hardware and, if possible, software procurements that provides: (a) rationale for defining risks, expected program costs for the Sneak Circuit Analyses reliability and maintainability enhancements, schedule impacts, and cost effectiveness for various equipments or system types complexities and environments, (b) information useful for determining, based on equipment complexity and other factors, the scope and depth of a Sneak Circuit Analysis required.

The basic data for this task have been developed in preceding tasks and encompass hardware and software systems. This task uses the previous task results as a basis for developing the Sneak Analysis Application Guidelines. Guidelines concerning expected program costs and analysis cost effectiveness required additional research and information gathering. Section 3.2 task results have already established the average number of equipment and software reports per environment. The additional required data was the cost by phase to correct these identified problems. The literature search resulted in the identification of a trend chart for software systems. No corresponding charts were found for hardware systems. Equivalent cost data to correct hardware system problems were derived from the information on the Short Range Attack Missile (SRAM) program. The results of this effort were cost trend charts which present the relative cost to correct Sneak Analysis identified problems. The hardware and software trend lines are similar.

A summary of the Application Guidelines is presented in Section 3.5.7.

3.5.1 Rationale for defining risks. The defining of risks from the procuring activity's perspective can be considered in two ways:

1. Program risk in not performing Sneak Analysis
2. Program risk in performing Sneak Analysis

The risks associated with not performing Sneak Analysis include the following:

1. Missed or overlooked problems - The effects of these problems can range from no man or mission effects to catastrophic effects. One of the primary findings resulting from the detailed study of SCA effectiveness is that significant levels of equipment/software problems are present in systems, regardless of the program development phase. Reference Tables 3-4 through 3-7 and Tables 3-28 through 3-31. Sneak Analysis identifies some of these program problems based on an analysis technique that assumes no equipment failures. Other failure related analyses can be combined with Sneak Analysis in the problem identification effort.

2. Increased Change Costs - Section 3.5.2 develops the cost figures for correcting problems by program phase. If Sneak Analysis is not performed, some or possibly all of the problems expected to be identified in the analysis may eventually be detected by other analyses, testing or event occurrence. Assuming that many of the same problems are found, the net effect will most likely be increased program incurred costs due to finding the problems in later development phases. Change costs in the Pilot and Unlimited Production phases can have a major impact on overall program funding allocations.

3. Schedule Impact - The occurrence of system problems in later development phases can delay the accomplishment of scheduled program milestones. Early detection of these problems is suggested to allow for an orderly redesign of the equipment/software system. This should reduce program schedule impacts associated with finding and correcting system problems.

4. Reduced Reliability/Safety - Any unidentified problems in the overall hardware/software system configuration are merely conditions awaiting the time and operating mode(s) to occur. The occurrence of the problems can be man or mission critical and can thereby adversely affect the planned program safety and reliability estimates. The criticality results illustrated in Tables 3-28 through 3-31 reflect the pronounced number of identified problems in Criticality I systems, the decreased number of problems in mission critical systems, and the lowest number of problems in non-mission critical systems. The occurrence of these problems and the potential for these problems to cascade through the system certainly decrease reliability and safety margins.

5. Greater Testing/Analysis Requirements - The Sneak Analysis results obtained especially in the Unlimited Production Phase are conclusive evidence that general testing and analysis requirements do not identify all system problems. There are limits to the capabilities of these functions. Sneak Analysis complements these functions and decreases anticipated test and analysis resource and cost allocations. If Sneak Analysis is not performed, increased reliance must be placed on system testing and other design and fault analyses.

6. National Prestige - This is a factor associated with catastrophic failure on one extreme and complete mission success on the other extreme. This factor is difficult to define or quantify. Certainly the program requirement to establish <u>necessary</u> and <u>sufficient</u> safety and reliability levels is <u>a guiding</u> policy, <u>but it must</u> be tempered with the consideration of the mission or program intent. The tools used to achieve these necessary and sufficient requirements should include Sneak Analysis, since it is one of the few tools available for performing an overall detailed systems analysis.

The risks associated with performing Sneak Analysis include the following:

1. Late Results - The analysis may be performed too late in the development phase to implement cost effective design corrections. In addition, identified problems may be resolved procedurally or rationalized as inconsequential rather than correcting the problems. The problems then remain in the system awaiting the correct circumstances for occurrence. Those problems that are fixed may not be subjected to the same degree of testing and analysis due to pressing program milestones, thereby compromising mission safety and reliability. Consideration should be given to performing the analysis in an early development phase with a continuing change analysis activity to prevent sneaks from being designed into the constantly changing baseline system.

2. Incorrect Scope and Depth of Analysis - If the Sneak Analysis effort is not properly scoped, the detailed tracking of system events and functions may be inhibited. The Sneak Analysis contractor can be put into a position of having to assume system operation and configuration. Identification of inadvertent system functions requires sufficient documentation to configure the complete equipment circuit and software routine functions. The procuring activity should take an active role in determining project scope. Implicit in this discussion is the requirement that the equipment drawings and software code listings represent the current and complete configuration. For the Validation and Conceptual phases, the configuration drawings or software logic diagrams will most likely be at the system and subsystem level, resulting in performance of a high level Sneak Analysis. From the Full-Scale Engineering Development Phase on to Unlimited Production, detailed drawings and code should be available to perform a more in-depth analysis. In general, Sneak Analysis tasks should be performed at the detailed component or instruction level, since many of the problems normally identified by the analysis would be missed if the depth requirement limited the analysis to higher level systems and subsystems. An inspection of report conditions contained in virtually any Sneak Analysis Final Report should provide justification for this position.

3. Assurance - Like other analysis tools, no absolute assurance can be given that all problems have been identified in the analyzed system(s). Cause and effect relationships within the system become more difficult to establish as the areas of analysis are scoped to smaller and smaller entities. Quality control checks on the analysis are critical for both large and small jobs.

3.5.2 Expected program costs. This section addresses those program costs which the procuring activity can expect to incur as a result of correcting Sneak Analysis identified problems. The approach is largely statistical in nature. The information represents a single program in the Airborne Environment, but one that may be considered a typical project application for Sneak Analysis. The program selected for this study is the Short Range Attack Missile (AGM-69A and AGM-69B). This program was selected for study because of the availability of information. Sneak Analysis was not performed on this program. The main design and production phases for this program occurred in the late 1960's and early 1970's when Sneak Analysis was largely confined to relay logic hardware designs on manned spacecraft vehicles.

Program funding for the SRAM program began in 1964 and continues to the present time period. Table 3-39 presents the overall annual program funding from 1964 to the present, along with individual sums for Research, Development, Test and Evaluation (RDT&E) and procurement.

TABLE 3-39. SRAM PROGRAM FUNDING

YEAR	R&D FUNDING (MILLIONS)	PROCUREMENT FUNDING (MILLIONS)	TOTAL FUNDING (MILLIONS)
1964-66	32.2	0	32.2
1967	30.8	0	30.8
1968	56.6	0	56.6
1969	83.3	16.1	99.4
1970	84.7	10.0	94.7
1971	56.5	112.9	169.4
1972	12.1	233.1	245.2
1973	0	195.2	195.2
1974	0	131.1	131.1
1975	0	0 (2.5 MODS)	2.5
1976	3.0	0 (2.5 MODS)	5.5
1977	1.1	0 (1.0 MODS)	2.1
1977	15.5	25.3 (AGM-69B)	40.8
1978	12.2	0	12.2
1979	8.9	0	8.9
1980	8.2	0	8.2
	$405.1	$729.7	$1134.8

A graphical representation of the Table 3-39 information is presented in Figure 3-8 and illustrates the two major peaks in program funding. The first peak is the 1969-1970 time period for the RDT&E portion of the SRAM program, followed by the 1971-1974 time period for procurement. A total of 1500 SRAM's were procured under this program effort.

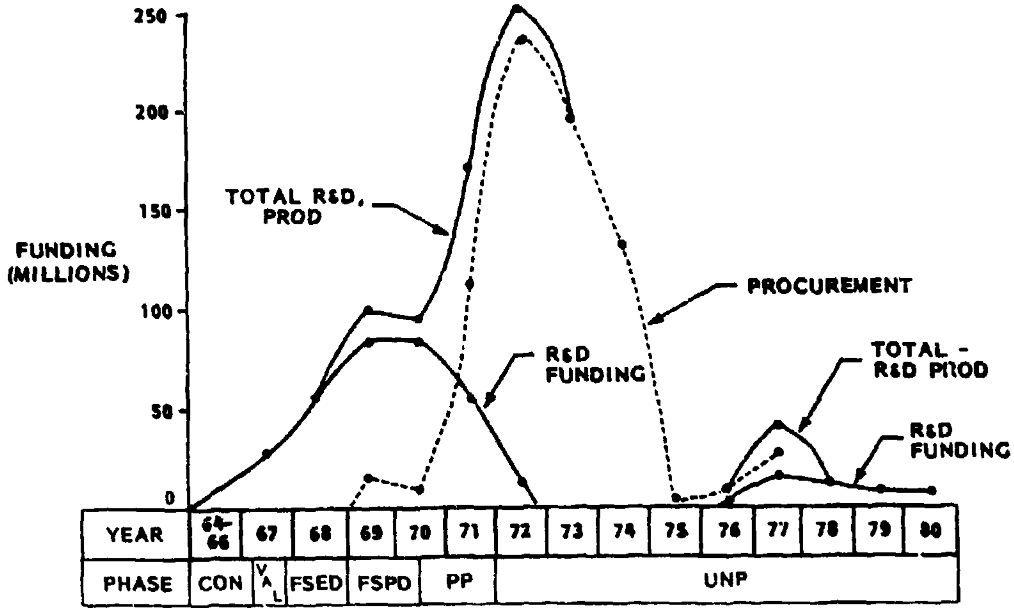


Figure 3-8. SRAM Program Funding

The next important block of information is displayed in Table 3-40, which represents the design history of change incorporation experienced on the SRAM program. The first major change activity occurred in 1967 and continued through 1974, the final year for missile procurement.

TABLE 3-40. SRAM DESIGN CHANGE HISTORY

YEAR	NUMBER OF DCN's	NUMBER OF ADCN's	TOTAL CHANGES	TOTAL PROGRAM COST ($M)	AVERAGE CHANGE COST $
1966	0	0	0	32.2	-
1967	400	900	1300	30.8	23,692
1968	1000	1700	2700	56.6	20,963
1969	900	1000	1900	99.4	52,316
1970	600	800	1400	94.7	67,142
1971	500	700	1200	169.4	141,167
1972	200	400	600	245.2	408,167
1973	150	200	350	195.2	557,714
1974	150	100	250	131.1	524,400

Table 3-40 presents the annual statistics for the number of Design Change Notices (DCN's) and Advanced Design Change Notices (ADCN's). A composite annual figure for all changes is also presented. For purposes of this study, it was assumed that all design (basic and changes) beginning in 1967 would be represented in one or more DCN's or ADCN's. No record of changes appears in the 1964 to 1966 time period, and it was assumed that no basic design was present in this time period. Under this approach, the development of the basic design and all subsequent changes to that design may be tracked to specific DCN's and ADCN's. Total program funding was then obtained from Table 3-39. Division of the total annual program costs by the total number of annual system changes resulted in an average annual cost per change. This approach spreads the entire cost of tooling, materials, acquisition, manufacturing, design and test, and modifications over all changes in the SRAM program. This is not a desirable approach, but it was the only approach available for this study. Actual tracking of change costs by change by year was not possible for this or other projects because of lack of data, as referenced in Section 1.3. Numerous changes are typically included in a single procurement, but actual resource costs to implement the changes are difficult if not impossible to track. A more realistic approach would provide an annual level of change cost from the initial contract award to contract termination. The trends identified by this assumed change cost distribution are presented in Figure 3-9.

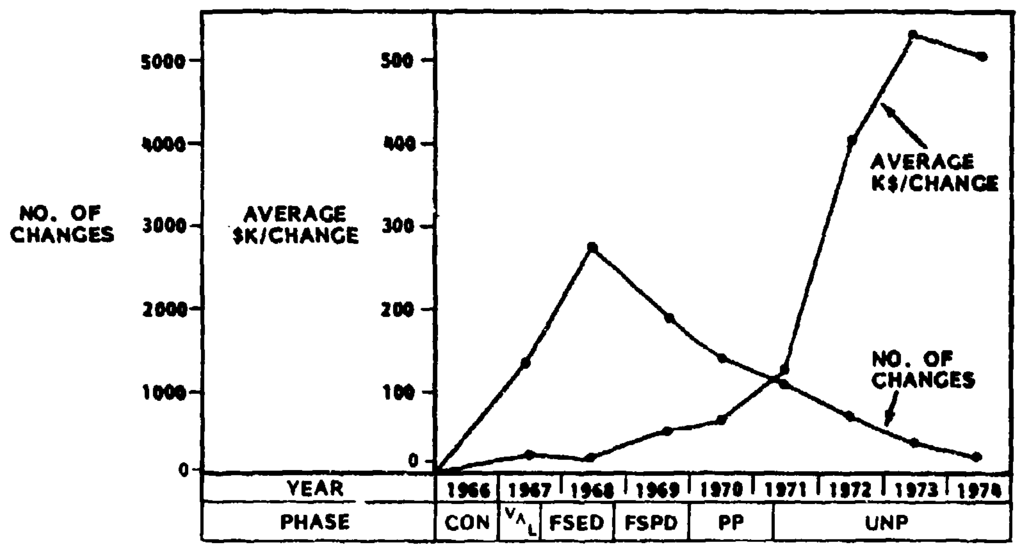


Figure 3-9. Number and Cost of SRAM Program Changes

Figure 3-9 depicts the change incorporation history for SRAM. Program changes began in 1967, peaked in 1968 and tapered to zero by 1974. This change history appears reasonable for the program duration shown. A sharper rise and fall might be expected for Space Environment programs which involve a very small number of vehicles, for example. The SRAM change levels are very smooth with no interim fluctuations other than the peak in 1968.

Figure 3-9 also illustrates the trend toward increased program cost to implement an "average" change. Once the procurement cycle is initiated, the average change cost increases dramatically.

The information in Figures 3-3 and 3-9 and Tables 3-39 and 3-40 represents the base information for determining a representative cost to apply for an average program change. The next step in the process of determining expected program costs will be to summarize the number of problems found in past Sneak Analysis projects. The product of the number of problems found and the average cost per problem summed to the cost of the Sneak Analysis represents the total program cost; that is, total program cost = (number of changes x cost per change) + Sneak Analysis cost. To generate a meaningful distribution of cost, dollars and number of problems are grouped by program development phase.

Table 3-41 is a partial reproduction of Table 3-6, which represents the average number of Airborne Environment Sneak Analysis Reports by program development phase. Only the figures for Sneak Circuit Reports (SCR's), Design Concern Reports (DCR's), Software Sneak Reports (SSR's), and Software Design Concern Reports (SDCR's) are presented. The Document Error Report (DER) categories have been omitted in Table 3-79 because they are concerned with discrepancies between the various configuration drawings, specifications, and other supporting documentation. In some instances, these discrepancies are actual representations of sneak conditions. However, since the majority of DER's do not result in sneaks and because of their high level of occurrence in a program, the report levels were not included in the following tables and figures.

TABLE 3-41. SNEAK ANALYSIS REPORT AVERAGES, AIRBORNE ENVIRONMENT

SNEAK ANALYSIS REPORTS / DEVELOPMENT PHASE	CONCEPT	VALID	FSED	FSPD	PP	UNLIM PROD	AVERAGE
SCR's	0	0	20	8	6	5	9
DCR's	0	0	15	12	5	7	10
HW SUBTOTALS	0	0	35	20	11	12	19.5
SSR's	0	0	2	3	17	7	10
SDCR's	0	0	8	9	20	12	14
SW SUBTOTALS	0	0	10	12	37	19	19.5

Table 3-42 is a cost summary of all Sneak Analysis projects by environment, with separate totals for hardware and software projects. The table represents the non-adjusted project cost averages and the adjusted project cost averages for a 1981 base. The non-adjusted average costs were obtained by summing all project costs within a particular classification and dividing by the number of

projects in that classification. The adjusted project costs were obtained by first converting all project costs to a 1981 cost basis, summing all projects within a particular classification and dividing by the number of projects in that classification. Table 3-43 was used in converting prior year costs to 1981 costs. Two numbers per category are presented in Table 3-42 for the Space Environment and the Composite Environments. The first number of the two-number set represents the total sample of Space Environment projects, including Apollo Skylab and Shuttle. The second number of the two-number set represents the sample of projects excluding the three large projects.

TABLE 3-42. ACTUAL AND ADJUSTED SNEAK ANALYSIS PROJECT COSTS

ENVIRONMENT ANALYSIS TYPE	NON-ADJUSTED AVERAGE ($K)	ADJUSTED 1981 AVERAGE ($K)
SPACE		
HARDWARE	614/82	1136/153
SOFTWARE	0	0
SUBTOTALS	614/82	1136/153
AIRBORNE		
HARDWARE	77	113
SOFTWARE	60	77
SUBTOTALS	73	106
GROUND		
HARDWARE	69	98
SOFTWARE	97	108
SUBTOTALS	75	100
COMPOSITE		
HARDWARE	177/75	302/115
SOFTWARE	75	89
TOTALS	163/75	275/111

TABLE 3-43. ADJUSTMENT FACTORS FOR PROGRAM AND PROJECT COSTS

YEAR	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981
PERCENT INCREASE	0	3	3	3	4	4	5	6	7	8	9.5	11	12.5	14	15
ADJUSTMENT FACTOR	2.724	2.645	2.567	2.492	2.396	2.305	2.195	2.071	1.936	1.792	1.637	1.475	1.311	1.150	1

Assembling all of the previous information into a single table allows a determination of eventual program costs. Table 3-44 represents a compressed program data set displaying the total cost to perform Sneak Analysis and the program cost to correct the Sneak Analysis identified problems.

TABLE 3-44. DERIVED HARDWARE PROGRAM CHANGE COST BY PHASE, AIRBORNE ENVIRONMENT

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	CHANGE COST	$113K ADJUSTED SA COST	TOTAL CHANGE COST	COST MULTIPLES
CONCEPT	0 *(10)	5,000	(50,000)	40,000	(90,000)	1X
VALIDATION	0 *(20)	20,000	(400,000)	41,000	(441,000)	5X
FSED	35	20,000	700,000	41,000	741,000	8X
FSPD	20	35,000	700,000	43,000	743,000	8X
PP	11	90,000	990,000	45,000	1,035,000	12X
UNLIMITED PRODUCTION	12	400,000	4,800,000	49,000	4,489,000	54X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

The SRAM program major milestones were presented in the Missiles and Spacecraft volume of the DMS Market Intelligence Report. These milestones were used to partition the SRAM information into the six program development phases. The time period for each phase may be determined from the program year shown in Table 3-44. The number of Sneak Analysis Reports is a direct extraction from the hardware portion of Table 3-41. Estimated report levels for the Concept and Validation phases were also included. The average cost per change was interpolated from Table 3-40 after adjustment for program phase. The next column of Table 3-44 is the Change Cost obtained by multiplying the number of changes by the cost per change. The column entitled $113K Adjusted Sneak Analysis Cost represents a readjustment of the 1981 average cost for an Airborne Environment hardware project to the applicable SRAM program development phases. This cost summed with the change cost represents the total program cost associated with identifying and correcting Sneak Analysis problems. Notice that in this table, Sneak Analysis Cost to Total Change Cost ranges from approximately 45% at the Concept phase to approximately 1% at the Unlimited Production phase.

A more graphic representation of costs is illustrated in Figure 3-10. This figure represents a relative cost curve based on the Table 3-44 results. Using the Concept phase dollar level as a basis, each succeeding program phase is represented as a cost multiple and plotted. The resulting trend shows a pronounced cost multiple in the Unlimited Production phase.

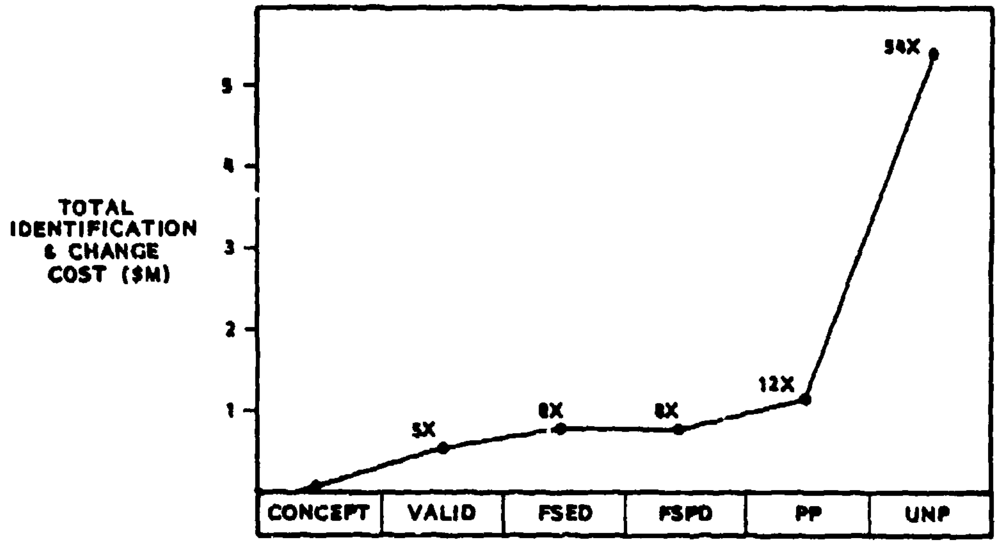


Figure 3-10. Relative Hardware Program Change Cost Trend, Airborne Environment

One additional interpretation can be obtained from Figure 3-10. The relative cost difference between any two phases can be determined. For example, the cost to identify and correct a problem at FSPD is eight times more expensive than at the Concept phase. The cost at the Unlimited Production phase is approximately seven times higher than at the FSPD phase.

The above tables and figures represent an Airborne Environment hardware program. Adopting the same approach, Tables 3-45 through 3-47 were derived for the Composite, Space and Ground/Water Environments. These tables also contain the relative cost multipliers.

TABLE 3-45. DERIVED HARDWARE PROGRAM CHANGE COST BY PHASE, COMPOSITE

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	$K CHANGE COST	$302/115K ADJUSTED SA COST	TOTAL COST $K	COST MULTIPLES
CONCEPT	0 *(10)	5,000	50	108/41	158/91	1X/1X
VALIDATION	0 *(20)	20,000	400	111/42	511/442	3X/5X
FSED	27,53	20,000	1060/540	111/42	1171/582	7X/6X
FSPD	23	35,000	805	114/43	919/848	6X/9X
PP	21	90,000	1890	121/46	2011/1936	13X/21X
UNLIMITED PRODUCTION	15	400,000	6000	131/50	6131/6050	39X/66X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

TABLE 3-46. DERIVED HARDWARE PROGRAM CHANGE COST BY PHASE, SPACE ENVIRONMENT

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	K CHANGE COST	$1136/153K ADJUSTED SA COST	TOTAL COST $K	COST MULTIPLES
CONCEPT	0*(10)	5,000	50	405/55	455/105	1X/1X
VALIDATION	0*(30)	20,000	600	417/56	1017/656	2X/6X
FSED	5/117	20,000	2340/100	417/56	2757/156	1X/6X
FSPD	55	35,000	1925	429/58	2354/1983	5X/19X
PP	41	90,000	3690	456/61	4156/3751	9X/36X
UNLIMITED PRODUCTION	15	400,000	6000	493/66	6493/6066	14X/58X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

TABLE 3-47. DERIVED HARDWARE PROGRAM CHANGE COST BY PHASE, GROUND/WATER ENVIRONMENT

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	CHANGE COST	$98K ADJUSTED SA COST	TOTAL CHANGE COST	COST MULTIPLES
CONCEPT	0*(10)	5,000	50,000	35,000	88,000	1X
VALIDATION	0*(20)	20,000	400,000	36,000	436,000	5X
FSED	26	20,000	520,000	36,000	556,000	6X
FSPD	27	35,000	945,000	37,000	982,000	11X
PP	12	90,000	1,080,000	39,000	1,119,000	13X
UNLIMITED PRODUCTION	20	400,000	8,000,000	43,000	8,043,000	91X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

Table 3-48 merges the total hardware program costs into one area for comparison purposes. The program costs for the Airborne and Ground/Water Environments compare rather closely, with the one exception occurring in the Unlimited Production phase. The Ground/Water Environment Change Cost for this phase is the highest of the sample set. The Space Environment starts with a high change cost and maintains this high differential cost throughout the development cycle.

TABLE 3-48. TOTAL HARDWARE CHANGE COSTS

PROGRAM ENVIRONMENT / DEVELOPMENT PHASE	COMPOSITE $K	SPACE $K	AIRBORNE $K	GROUND/WATER $K
CONCEPT	158	455	90	88
VALIDATION	511	1,017	441	436
FSED	1,171	2,757	741	556
FSPD	919	2,354	743	982
PP	2,011	4,146	1,035	1,119
UNLIMITED PRODUCTION	6,131	6,493	4,849	8,043

Using the same approach as above, Tables 3-49 through 3-51 were derived for software projects for the Composite, Airborne and Ground/Water Environments. The Space Environment has no past software projects on which to derive a basis, although several new projects have been initiated since the start of the RADC Application Guidelines effort. In general, the software program change costs rise at a slower rate in the early development phases and exceed the hardware costs in the latter two development phases. Software change cost trends indicate a greater penalty associated with late identification and correction of program problems.

TABLE 3-49. DERIVED SOFTWARE PROGRAM CHANGE COST BY PHASE, COMPOSITE

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	CHANGE COST	$89K ADJUSTED SA COST	TOTAL CHANGE COST	COST MULTIPLES
CONCEPT	0*(5)	5,000	25,000	32,000	57,000	1X
VALIDATION	0*(10)	20,000	200,000	33,000	233,000	4X
FSED	15	20,000	300,000	33,000	333,000	6X
FSPD	12	35,000	420,000	34,000	454,000	8X
PP	41	90,000	3,690,000	36,000	3,726,000	65X
UNLIMITED PRODUCTION	19	400,000	7,600,000	39,000	7,639,000	134X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

TABLE 3-50. DERIVED SOFTWARE PROGRAM CHANGE COST BY PHASE, AIRBORNE ENVIRONMENT

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	CHANGE COST	$77K ADJUSTED SA COST	TOTAL CHANGE COST	COST MULTIPLES
CONCEPT	0*(5)	5,000	25,000	27,000	52,000	1X
VALIDATION	0*(10)	20,000	200,000	28,000	228,000	4X
FSED	10	20,000	200,000	28,000	228,000	4X
FSPD	12	35,000	420,000	29,000	449,000	9X
PP	37	90,000	3,330,000	31,000	3,361,000	65X
UNLIMITED PRODUCTION	19	400,000	7,600,000	33,000	7,633,000	147X

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

TABLE 3-51. DERIVED SOFTWARE PROGRAM CHANGE COST BY PHASE, GROUND/WATER ENVIRONMENT

PHASE	NO. OF SA CHANGES	**COST/ CHANGE	CHANGE COST	$108K ADJUSTED SA COST	TOTAL CHANGE COST	COST MULTIPLES
CONCEPT	0*(5)	5,000	25,000	39,000	64,000	1X
VALIDATION	0*(10)	20,000	200,000	40,000	240,000	4X
FSED	17	20,000	340,000	40,000	380,000	6X
FSPD	0	35,000	0	41,000	-	
PP	51	90,000	4,590,000	43,000	4,633,000	72X
UNLIMITED PRODUCTION	0	400,000	0	47,000		

* ASSUMED REPORT LEVELS
** FIGURE 3-9 LOOKUP

Table 3-52 merges the total software program costs into one area for comparison purposes. A detailed comparison, however, may provide erroneous trends considering the relatively small number of software projects.

TABLE 3-52. TOTAL SOFTWARE PROGRAM CHANGE COSTS

DEVELOPMENT PHASE \ PROGRAM ENVIRONMENT	COMPOSITE $K	SPACE $K	AIRBORNE $K	GROUND/ WATER $K
CONCEPT	57	-	52	64
VALIDATION	233	-	228	240
FSED	233	-	228	380
FSPD	454	-	449	-
PP	3,276	-	3,361	4,633
UNLIMITED PRODUCTION	7,639	-	7,633	-

During the literature search for change costs, relative cost levels to correct software problems were identified in the July 23, 1981 Electronic Design Magazine, page 75. The base reference is "Tutorial-Software Design Strategies," IEEE, (EHD 149-5), 1974; authors Glen D. Bergland and Ronald D. Gordon. These relative costs are displayed in Figure 3-11. A reasonably close approximation exists between the relative software program costs shown in Tables 3-49, 3-50, and 3-51. The one major difference occurs in the Test or Pilot Production phase, and that difference may be caused primarily by the selection of program phase boundaries shown in Figure 3-11.

3.5.3 Cost-Effectivity. Sneak Analysis cost-effectivity can be measured in terms of return on investment and cost avoidance. The return on investment for Sneak Analysis projects in terms of program dollars spent to number of report conditions is typically greater when performed in the early development phases. The distribution of hardware Sneak Analysis costs through the various program development phases is uniform, as shown in Table 3-15. The number of reports generated by phase is shown in Figure 3-12. The figure illustrates the higher number of hardware reports found in the early development phases. Thus, for hardware systems, more reports are generated in the early development phases for approximately the same program dollar expenditure.

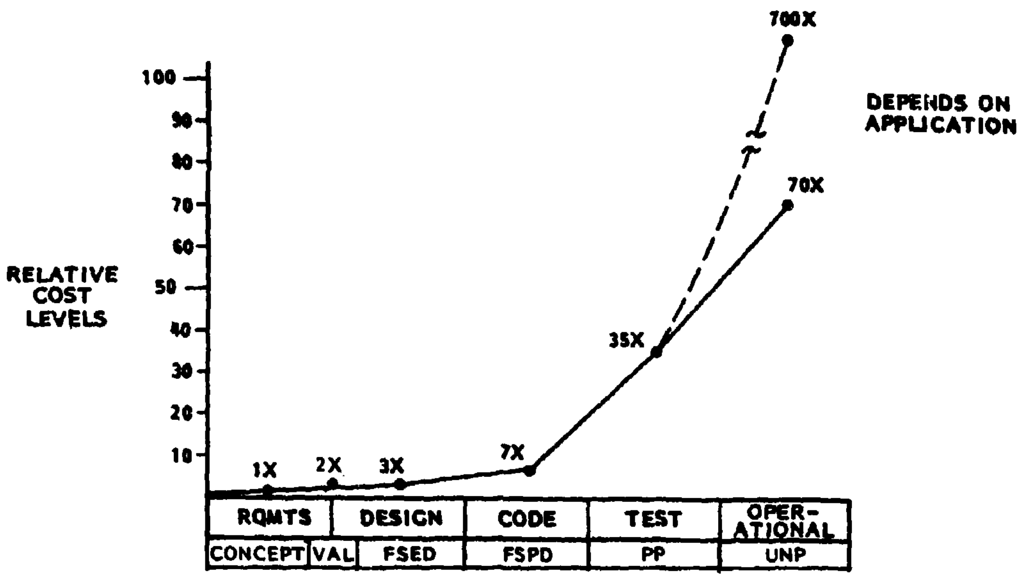


Figure 3-11. Relative Software Program Change Costs

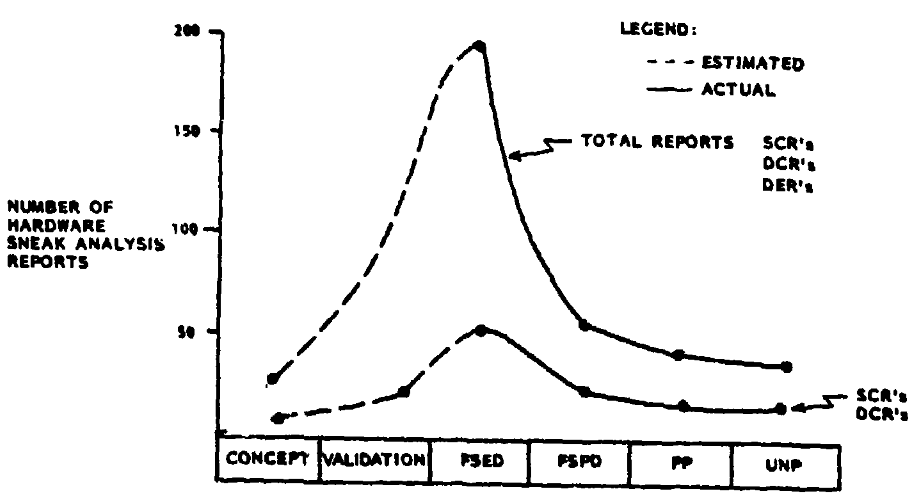


Figure 3-12. Development Phase Distribution of Hardware Reports

The distribution of software Sneak Analysis costs through the various program development phases illustrates low average costs in the early phases and higher costs in the later phases. The distribution of software reports by phase is shown in Figure 3-13 and differs noticeably from the hardware trends. The greatest number of reports occurs in the Pilot Production phase, two phases later than the hardware trend. The offset is produced by the small number of software projects and by project reliance on completed software code.

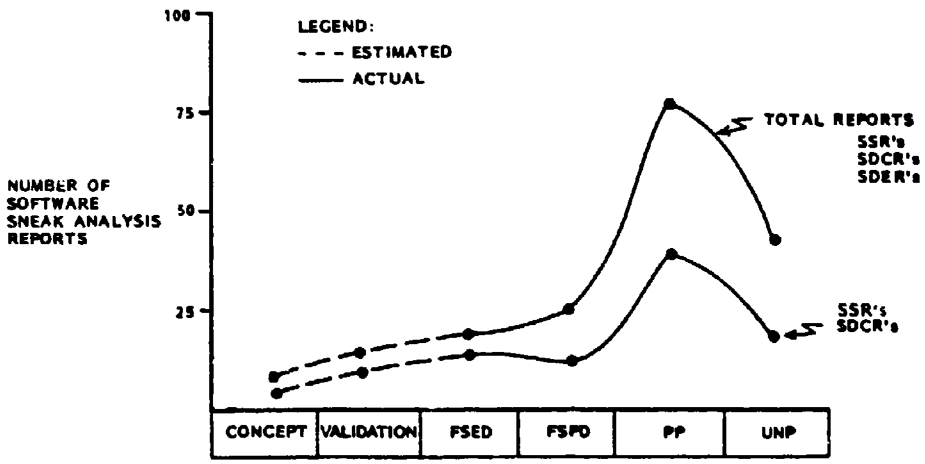


Figure 3-13. Development Phase Distribution of Software Reports

If Sneak Analysis becomes a formal part of Verification and Validation (V&V) efforts, the early development phase report levels should increase, effectively moving the report peak to an earlier phase. With low costs and higher average report levels in the earlier phases and higher costs and high average report levels in the latter phases, there is an evident need to direct the use of software Sneak Analysis to earlier development phases to obtain maximum cost benefit. The analysis would have to be implemented on each of the developed program modules to identify modular level problems earlier and then followed by analysis of the complete program code. If the analysis is implemented at an early development phase, it is suggested that the Change Analysis option be selected to compensate for the relatively large number of problems identified in the later development phases. If software Sneak Analysis is not performed in the early development phases, it should be used in the later phases because of the relatively high number of problems which are identified.

Note: Cost-effectivity based on averages can be misleading especially in the context of overall program costs. Meeting or exceeding the average report levels is not a prime consideration or expected result for any particular project application. The identification of embedded system problems is the primary consideration, regardless of the numbers found. The identification of only one serious hardware or software condition can more than offset the cost of the analysis, and in the case of the later development phase, there can be an actual cost savings. In addition, problems that cannot be and were not found by other analysis tools can also be identified in a cost-efficient manner.

The F-4C Nose Wheel Steering Sneak Analysis project is one such example of an embedded problem which was identified for a small program expenditure. The steering problem existed in the fleet of aircraft and had escaped detection by other means, until found in the Sneak Analysis effort. The analysis was undertaken to find only one major system problem, which it did identify.

An additional return on investment occurs in the Pilot and Unlimited Production phases for combined analyses. Based on analysis of the Appendix A Project History Table data, a majority of projects which combined Sneak Analysis with other analysis tools (FMEA's, FTA's, PHA's, etc.) are present in the latter two development phases. The resultant report levels for just the Sneak Analysis portion compare very closely to the report levels shown in Figure 3-12. The projects costs for the combined analyses are slightly higher than Sneak Analysis alone. In effect, more analysis is produced for the given program investment. A suggested approach is to perform Sneak Analysis along with hazard analyses (PHA's, SHA's, SSHA's, etc.) at an early development phase such as FSED and follow this effort with additional analyses such as FMEA's, FTA's, Common Cause Failure Analysis, and Single Point Failure Analysis. The initial hazard analyses identified the primary system and subsystem level problems; Sneak Analysis produces the network tree configuration and identifies detailed system problems which are not fault dependent; and the final analyses search for fault related problems and system effects. One major cost reduction associated with combined analyses is the elimination of redundant documentation acquisition efforts and redundant determination of system configurations. Some of the redundant analysis and problem reporting functions are also eliminated in combined analyses.

Cost avoidance accrues to the program when problems are identified and corrected early in the development cycle before cost becomes a dominant factor. Tables 3-44 through 3-51 display a numeric cost multiplier which can be used to determine cost savings. The cost multiplier establishes the relative cost to perform Sneak Analysis and implement corrective fixes to system problems.

A base level of one is established for the concept phase. The multiplier numbers are then increases in costs over the concept phase costs. Differences in program costs between any two phases represent added or avoided costs. Using the Airborne Environment Table 3-44, the cost multiplier for performing hardware Sneak Analysis and incorporating system corrections in the FSED phase is 8 X ($741K), while the same effort in the Unlimited Production phase is 54 X ($4,489K). The cost avoidance for FSED is then approximately a factor of 6 X ($3,748K) of the Unlimited Production phase analysis.

The Ground/Water Environment Table 3-47 contains the largest hardware cost spread of the three environments. This distribution of cost is produced by the relatively high number of reports identified in the later development phases. A cost differential of 15 X occurs between the FSED and Unlimited Production phases.

Figure 3-11 illustrates a greater differential between phases for software programs. The differential between the FSED and Unlimited Production phases is 25X to 250X. The higher multiple can be expected and is reasonable when software development is not firmly rooted and guided by the requirements and design phases. Programs with formal V&V should tend toward the lower relative cost level. Figures 3-11 and 3-13 demonstrate the importance of problem identification in the later phases. Regardless of the testing and approach, reportable problems still exist. Maximum cost avoidance for software development can be achieved if Sneak Analysis is performed at the FSED while the design is still fluid and can be changed for relatively small cost.

3.5.4 Schedule impacts. Schedule impacts can occur at the program level and/or at the Sneak Analysis Project level. The problems identified in Sneak Analysis reports can and have produced program level schedule impacts. Report conditions identified in late development phases, in particular the Unlimited Production phase, can have serious implications for scheduled program milestones. Several Space Environment programs implemented hardware corrections while the vehicle was either on the launch pad or in a vehicle assembly building awaiting rollout to the launch pad. When these serious problems were found, potential design modifications were considered and then a final design fix was chosen and implemented. The extensive testing and consideration of undesirable operating modes were accomplished in a very short period of time, a situation conducive to error. Obviously, it would have been desirable to identify and correct the design or operating deficiencies at an earlier phase and allow for more extensive testing of the design corrections.

Since the output of a Sneak Analysis Project is the Sneak Analysis report set (SCR, DCR, DER, SSR, SDCR, SDER), the resultant program schedule impact is dependent on the number and types of problems identified. Numerous equipment or software conditions validated by the procuring activity and prime contractors can become an indicator that the systems or subsystems included in the analysis contain undesirable operating modes relevant to the safety of the crew and successful mission completion. Suggestions for correcting the design or operational deficiencies included in the report offer the problem review board at least one viewpoint for consideration. If the report conditions occur in areas previously analyzed, then consideration should be given to expand the analysis for the complete systems or subsystems.

The types of problems found in a Sneak Analysis can also have an impact on program schedules. One of the more desirable situations is to run a controlled test to demonstrate the problem and its consequent system effects. Some problems are of this category and are relatively easy to demonstrate. Other problems require repeated cycling in order for the condition to occur. Normally these equipment or software problems are associated with race conditions or equipment sizing and may only occur after some component degradation. These apparently non-repeatable or non-occurring problems give rise to the likelihood of occurrence argument. The reported condition may then be dismissed solely on the basis of a value judgment. Extreme care should be exercised in dismissing the report conditions and the possible system effects.

In some cases, a single Sneak Analysis report may document many conditions, although only one condition is described in the body of the report. For example, multiple points in a design intended to have redundancy may be compromised by a common condition, such as common connectors, power supplies, or electrical isolation. Rather than issue separate reports for each condition, only one report is issued and all conditions are listed. The resolution of the report conditions may be so extensive that only a major program redesign and/or consequent schedule slide are possible.

Once problems identified by Sneak Analysis are determined to have a schedule impact, proposed equipment or software modifications should be sent to the performing Sneak Analysis contractor. The Sneak Analysis contractor should then incorporate the proposed changes into the configuration baseline (network trees) and reanalyze the affected circuitry or software. The intent here is to assure that the modification achieves the desired system operation, and that no additional sneak conditions are generated. The analysis can be performed on a proposed modification, thereby saving schedule time in the redesign, implementation and testing cycle. The later the program development phase, the more the change analysis option should be considered by the procuring activity.

Schedule impacts can also occur in the performance of the Sneak Analysis project itself. The data acquisition phase of a Sneak Analysis project is critical to successful completion of the task. Timely data acquisition of the correct type documentation is required so that inspection of the data for completeness and hookup can be performed. Any missing data areas or incompatible configurations, especially at equipment or module interfaces, can be investigated and new drawings or source computer code program listings ordered. In general all data should be received within the first 20% of the Sneak Analysis project, but this may vary depending on the requirements of the particular application. In no case should the data acquisition schedule exceed one-third of the project duration, unless this data is change information. Failure to acquire data in a timely manner can result in task schedule slides and increased analysis cost.

All Sneak Analysis reports should be evaluated by the procuring activity in a timely fashion. Any comments on the reports, including final report dispositioning, should be transmitted to the Sneak Analysis contractor. This is especially important in the initial project phase for the Drawing Error Reports

for hardware and software. Discrepancies between various types of documentation are reported, including the stated assumption on the part of the contractor as to which document(s) is correct. If the assumed correction is incorrect, errors can occur in the subsequent network tree configurations, invalidating some downstream analysis. All drawing error conditions should be resolved and coordinated with the Sneak Analysis contractor prior to the first one-third of the project schedule.

No evident project schedule impacts occur as a result of the type of equipment or software to be analyzed. Digital type systems require a slightly longer performance schedule than relay type logic only because of the amount of circuitry incorporated in the system.

3.5.5 Analysis scope and depth. Scoping factors which should be considered in a Sneak Analysis application include the following:

1. The number and type of components and/or computer instructions
2. Schedule requirements
3. Data availability
4. Change Analysis
5. Estimated project cost

The most important scoping factor in a Sneak Analysis application is the number and type of components involved in the system or systems to be analyzed. The greater the number of components, the higher the cost and the longer the required project schedule. The type of equipment may be digital, analog, or relay for hardware systems, and high order or assembly language for software systems. No particularly significant scoping requirements are associated with any of these system types. Digital systems incorporate much more circuitry in small physical dimensions, but the analysis is basically performed using the same approach, changing only some of the Sneak Analysis clues. Relay and digital systems are the two most prevalent hardware categories selected for Sneak Analysis, as shown in Tables 3-8, 3-17, and 3-26. Hardware systems currently are tending more toward digital type applications. Assembly language is the most prevalent software category selected for Sneak Analysis. However, this software trend should change as more and more program applications shift to the high order languages.

Sneak Analysis project applications involve one or more of these equipment/software types, even though the scoping requirements may restrict the analysis to a particular system or subsystem. Table 3-53 displays a condensed listing of the equipment/software applications found in the more detailed Appendix A Sneak Circuit Analysis Project History Tables. The contractor selected for the analysis should possess the demonstrated capability to perform in these areas.

Some of the program applications have involved the entire vehicle or module, while other applications have been restricted to particular systems and subsystems. Program cost and schedule considerations late in the development cycle tend to limit the application of Sneak Analysis to particular systems and subsystems. The limitation is due to the lack of programmed or allocated funds for the analysis. The average 4-6 month analysis period of performance may not

support program milestones if not properly scheduled. If the targeted program represents a fielded system, there may be a problem concerning maintained and up-to-date drawings to use in the analysis. Initiation of Sneak Analysis in the earlier development phases allows for the analysis of more systems. The analysis can also be staggered to concentrate on particular systems first and then to initiate lower priority systems.

TABLE 3-53. SNEAK ANALYSIS EQUIPMENT/SOFTWARE APPLICATIONS

SNEAK ANALYSIS EQUIPMENT/SOFTWARE APPLICATIONS	
ELECTRICAL POWER GENERATION ELECTRICAL POWER DISTRIBUTION ELECTRICAL POWER CONTROL ELECTRICAL SUPPORT EQUIPMENT FLIGHT/LAUNCH SEQUENCER FLIGHT CONTROL SYSTEM GUIDANCE/NAVIGATION SYSTEM LANDING SYSTEM ATTITUDE CONTROL SYSTEM ANGLE-OF-ATTACK TRANSMITTER PROPULSION SYSTEM ENGINE CONTROL SYSTEM THRUST REVERSER/FULL CONTROL AVIONICS SYSTEM THERMAL CONTROL ORDNANCE/PYROTECHNIC SYSTEM ARMING AND FUSING SYSTEM WEAPON CONTROL SYSTEM FIRE CONTROL RADAR SYSTEM DETECTOR SYSTEM	CAUTION/WARNING SYSTEM MONITOR/CONTROL SYSTEM INSTRUMENTATION SYSTEM DATA ACQUISITION AND PROCESSING TELEMETRY/SIGNAL CONDITIONING DATA RECORDER COUNTERMEASURE/DISPENSER SYSTEM LASER SEEKER SYSTEM ENVIRONMENTAL CONTROL SYSTEM EJECTION SEAT SEQUENCER DIAGNOSTIC SYSTEM LIGHTING SYSTEM SAFETY SYSTEM TRANSPORTATION SYSTEM EXPERIMENT SYSTEM COMPUTER DATA LINK CONTROLLER SHOP TEST EQUIPMENT BLOWOUT PROTECTION SYSTEM TOWER LOWERING SYSTEM ON-BOARD SOFTWARE

A prime consideration in scoping a Sneak Analysis project is the completeness of functions depicted in the drawings and/or source program listings. Much of a system design may be included in the supplied documentation, but the requirement for documentation outside of the scoped area may be required. For example, assume that a flight control system interfaces with the electrical power system, navigation system, active control system and the computer system. If the equipment comprising only the flight control system is selected, undefined interfaces exist beyond which the performing Sneak Analysis contractor cannot see. Assumptions as to function, timing and configuration must be made in the course of the analysis. As much as possible assumptions should be based on interface documentation, which, while not showing the actual circuit or code detail, specifies function and pin or channel number assignments. It would be desirable to include documentation depicting the circuit/software configuration, eliminating the need to make assumptions.

Sneak Analysis task schedule requirements can also have an impact on task scoping. If the analysis effort is to coincide with and support major program milestones such as PDR, CDR, test, and first flight, then consideration needs to be given to the anticipated task duration. A majority of past projects indicate a task duration of from three to nine months, as shown in Table 3-16, with an overall project average of six months.

The scoping and scheduling of Sneak Analysis is dependent on the availability of system data. Complete data packages describing an individual system or subsystem should be available for the analysis. If the complete data is not currently available, the analysis can be scoped to those areas that are complete and the analysis sequenced into the remaining areas as data becomes available. In this way, the complete system or subsystem can be analyzed a section at a time and then the various sections joined in an integration analysis. If many desired functions are missing in the designated system and will not be available by at least one-third of the project duration, the procurement for this application should be slid until the documentation becomes available.

The performance of change analysis is very dependent on project duration. Short duration projects under three months are not ideal applications for change analysis, unless all baseline data and the complete change package are available at project initiation. Attendant delays in transmittal of data can impact short duration projects. Projects of six or more months duration are better candidates for change analysis. Formal change analysis is an extra cost option for Sneak Analysis tasks. Approved and released changes can be evaluated under this option. Proposed changes occurring in response to reported sneak conditions can and should be evaluated within the scope of the baseline Sneak Analysis effort.

Estimated project cost can be a major consideration in the performance of Sneak Analysis. Costing information provided in Appendix B is an indicator of analysis cost based on the types of equipment composition and the number of components/instructions. Actual cost may be significantly less when the task is scoped to analysis of particular system functions. Corresponding program dollars should be allocated and maintained in the reliability program plan so that the analysis can be performed. If the anticipated scope of the analysis and/or the systems to be analyzed change sufficiently, cost considerations may dictate a rescoping of the task. The cost tables should be used as an estimating tool to determine the extent of the systems that can be included. Digital and analog systems are higher cost due to the large amount of circuitry involved. High order language software applications typically cost more than an equivalent number of assembly language program applications. Additional tracking and accounting of functions is required in the high order language application.

Depth of analysis should be at the detailed component/instruction level, instead of at the subsystem or system level. Other analyses are available for considering the higher level analyses, but very few of the analyses can be implemented on the detailed level. Sneak Analysis is unique in approach and has demonstrated the capability to identify problems not found by these other analyses and extensive testing. The experience to date which is reflected in Figures 3-12 and 3-13 indicate that the detailed level is the desired level to perform Sneak Analysis. Some project appliations were performed at a higher level with good results, but additional problems could have only been found at the detailed level.

3.5.6 Application guideline summary. This section summarizes the Sneak Analysis Application guidelines which have been developed throughout this document. The primary guidelines include:

1. Establishing need for Sneak Analysis

   a. Reliability improvements in the overall program result from the identification and resolutions of system problems. Sneak Analysis is very effective in identifying problems which may be missed by other analyses.

   b. Independent analysis is currently the only established approach for the analysis. The analysis must be performed by a contractor independent of the design group to preserve the integrity of the effort. It is also an excellent analysis tool which can be used to verify or cross-correlate the results or findings of other analyses.

   c. Problem detection to eliminate the need for costly retrofits or redesigns in mass-produced systems and possible loss of irreplaceable one-of-a-kind systems such as spacecrafts or particular airborne equipment are immediate considerations for performing the analysis.

   d. High criticality of the systems to be analyzed also warrants the analysis. Man or mission critical systems are the most likely candidates. Low criticality systems may be eliminated from consideration as long as no active control functions are performed in these systems.

   e. Unresolved system problems that have not been found by other analyses or tests are also good candidates for Sneak Analysis. If the analysis is undertaken to identify or isolate these system problems (typically during late development phases) allow the contractor some additional leeway in cost and equipment/instructions included as in-scope. Frequently, the unidentified problem causes are located in "unrelated" equipment/software areas. Analysis of only problem prone functions, or areas where the problem is manifest, such as an instrument panel or test equipment, may be insufficient to locate the cause of the system problems.

   f. A high change rate in the baseline design can also be used to justify the analysis. Loss of the design configuration baseline resulting from greater than expected change

rates can be rectified by the detailed analysis of each change before the change is implemented in the hardware or software system.

g. Sneak Analysis is a cost-effective tool in all phases of program development, but the analysis results exhibit a pronounced effectiveness in early development phases, and particularly in the Full-Scale Engineering Development Phase.

2. Determining applicable systems

a. Systems which perform active functions are the primary candidates for Sneak Analysis. Electrical power, distribution and controls have traditionally been the main areas for hardware analysis. Computer programs which actively control and sequence system functions are good software candidates. In general, those C^3 systems which occur in the command and control areas are the primary candidates. Non-repairable systems are especially good candidates.

b. Passive systems that do not affect the overall program operation can be omitted from analysis consideration. This can include certain communication systems and navigating systems, such as stand-alone radars. Fire control radars, however, are integrated with other systems and provide direct control over specific functions. They are not passive systems. Highly redundant passive systems may also be excluded. Redundancy in control areas, however, is not grounds for eliminating the analysis. There may be design problems which compromise or destroy the redundant design.

c. Sneak Analysis can and has been successfully implemented on complete vehicle or program applications, as well as limited subsystem or functional applications. The analysis is best performed on configurations involving numerous system interfaces and large size systems. The high number of interfaces as well as the complex designs are primary causes of embedded sneak conditions.

d. The applicable systems should be completely specified by component or instruction level documentation in the form of schematics, drawings, wire lists and source computer program code so that the analysis can be conducted at the "as-built" and "as-coded" levels, respectively.

e. Detailed analysis of critical systems can be performed by blending various analysis techniques which bring to bear the best features of each analysis in identifying design and fault related problems. Favorable project results and costs are obtained in blended analyses. Highly critical functions can be identified by other high level system analyses such as a Preliminary Hazard Analysis or System Hazard Analysis.

3. Calculating project cost and allocation of program budget

a. The cost of Sneak Analysis can be computed on the basis of the number and type of hardware components and the number and type of computer program language instructions. The Appendix B cost tables are used in cost computations and assume the performance of a detailed Sneak Analysis for all of the components in the estimate.

b. Limited budgets may force scope reductions and restrict a broad program application of Sneak Analysis. The analysis can and has been scoped to individual systems, subsystems and functions. Excessive scoping, however, could limit the analysis effectiveness by eliminating the detailed function tracking which is typically developed across system boundaries. Acceptable project costs are possible by selection of limited program systems as illustrated in the Appendix A Project History Tables.

c. If program funding and/or documentation are major factors restricting performance of Sneak Analysis, then an incremental contracting approach can be undertaken. Perform Sneak Analysis on one or more of the higher criticality systems for which documentation is readily available. In a following fiscal period, contract for Sneak Analysis on the remaining systems, with the stipulation that the analysis includes the new systems and interfaces into the previously analyzed systems. This approach is especially desirable when detailed drawing or code instructions are missing for particular equipment or program modules, respectively. Functional diagrams should be made available to the Sneak Analysis contractor for these missing areas.

d. The procuring activity can expect annual program costs for Sneak Analysis and problem resolution to range from 0.1% in the early development phases to approximately 5% in the later phases. There are significant cost and risk penalties associated with late identification and resolution of system problems.

e. The ratio of Sneak Analysis cost to total change cost ranges from approximately 50% in the Concept phase to 0.5% in the Unlimited Production phase.

f. The percentage of Sneak Analysis cost for the entire program duration averages approximately 0.06% for each of the three program environments, with the highest level of 0.4% and the lowest level at 0.0001%.

g. Space Environment correction costs are the highest overall for the three environments, while the Ground/Water Environment has the highest single phase correction cost during Unlimited Production.

h. Program budget for the analysis should be allocated in the formulation of the reliability program plan and maintained throughout the development cycle for the desired schedule start time. If program dollars have not been programmed for the analysis, they may not be available when required.

i. Since Sneak Analysis can be effectively blended with other analyses, reduced project costs for the combined analyses can be achieved.

4. Scheduling requirements

a. Sneak Analysis should be scheduled so that final project results are obtained and can be adequately evaluated by the procuring activity and equipment manufacturers prior to the end of the Full-Scale Prototype Development Phase. Program costs to implement system changes increase dramatically after this phase, as shown in Figures 3-10 and 3-11.

b. The preferred start time is prior to CDR in the Full-Scale Engineering Development phase. This is an ideal time to provide a formal input into the design review process. Optional change analysis should be considered to track and evaluate the resulting system changes brought about by CDR.

c. Timely results can be obtained for all scheduled Sneak Analysis projects and also for those projects which are intended to identify a single test, operational, or fleet problem. For single problem oriented Sneak Analysis limited system scoping and available documentation can provide project results as soon as one to two months into the project schedule.

d. Orderly scheduling of Sneak Analysis can be based on the average four to six month project duration. Targeting the analysis to a specific program milestone can be performed by moving the start date back the specified number of calendar months. The two most important items affecting successful performance by the designated program milestone date are data availability and contractor performance.

5. Establishing contract requirements

a. Specification requirements are available in this document for Sneak Analysis. Reference Section 3.4.1 and Appendix I.

b. Request for proposal considerations have been presented in Section 3.4.2.1, which identify and describe the various tasks involved in the Sneak Analysis process. These items are intended to provide the procuring activity with necessary and sufficient project requirements. Since no formal documentation of the technique is available in open literature, these considerations are quite important in competitive and sole-source applications. The fundamental analysis approach is based on the systematic network tree technique which has served effectively in problem identification.

c. Evaluation criteria are provided in Section 3.4.2.2, which should aid the procuring activity in evaluating contractor responses to the RFP's and eventually selecting the contractor to perform the analysis. Important criteria are applicable contractor experience, intended approach, depth of analysis, and cost. Selection of an independent contractor to perform the analysis is preferred. Security requirements may be necessary for contractor personnel and facilities.

d. A majority (84%) of the Sneak Analysis projects have been awarded as sole source Firm Fixed Price contracts. Cost-Plus-Fixed-Fee contracts are awarded for long duration Sneak Analysis projects, large system analyses, and those projects with optional change analysis.

6. Procuring activity monitoring guidelines

a. Data acquisition has customarily been assigned to the procuring activity. If data acquisition is assigned to the Sneak Analysis contractor, extra cost is incurred. Proprietary data from vendors and contractors typically requires proprietary data agreements which may require significant time to acquire. Projects involving classified data require special data handling procedures and appropriate level security clearances for personnel and facilities.

b. Sneak Analysis report evaluation and coordination at problem review boards and engineering change boards are an important procuring activity function. Tracking all reports and their eventual dispositions is an important element in assuring effective program benefits for the project expenditure. The resolution of the identified problems provides a measure of reliability improvement and sneak finding capability of the performing contractor. Document error reports are typically found early in the project schedule, and once the network tree drawings or diagrams are generated, the primary re. rts are design and sneak condition reports.

c. Liaison, contract monitoring, contract modification and project closeout are the remaining procuring activity functions.

3.6 Task 6 - Feasibility Study. Task 6 - The contractor shall perform a feasibility study on developing simplified or modified Sneak Circuit Analysis techniques that are applicable to small scale (i.e., part complexity of less than 5000) or one of a kind equipments. The study shall investigate schedule impacts, program costs and sneak analysis effectiveness.

3.6.1 Special considerations. The number of components and mix of components in a system have a special bearing on this feasibility study. The basic feasibility study is intended to be sized for a system of 5000 or less components. The intended system sizing limit is inordinately large, however, when the system composition is considered. Table 3-54 has been compiled using selected system size values and a cost determination made by use of Table B-1 of Appendix B. The values are based on a detailed Sneak Analysis of all components in the system for the specified component mix. The values would be significantly less when extraneous circuitry beyond designated system functions are excluded from the analysis. Scoping of the task to specified functions permits the application of Sneak Analysis to large systems for moderate cost as shown by a review of the Appendix A projects.

The threshold limit of 5000 hardware components for typical mixes of relay logic, general systems and highly digital logic systems results in an analysis cost of $400,000 or more. Only 4% of the 109 Sneak Analysis projects have occurred in this range, as shown in Table 3-13. That table is based on actual Sneak Analysis costs. If the actual Sneak Analysis cost is adjusted to a 1981 dollar basis by use of Table 3-43, less than 7% of the projects occur in this sizing category. Note that the component mix has an important effect on cost for this particular level.

When the average Sneak Analysis project cost is considered, the adjusted 1981 dollar cost is $111,000, as shown in Table 3-42. This cost average is high, because some of the past projects included not only Sneak Analysis but additional analyses such as FMEA's, FTA, etc. At this average project cost level, the equivalent number of components analyzed in detail for a relay logic mix would be 1405; 1181 components for the generalized mix, and 391 devices for the MSI mix. These levels are significantly below the desired 5000 component threshold.

TABLE 3-54. HARDWARE SNEAK ANALYSIS COST ESTIMATES BASED ON NUMBER AND MIX OF COMPONENTS

NUMBER OF HARDWARE COMPONENTS	RELAY LOGIC MIX AT $79/COMPONENT ±14%	GENERALIZED MIX AT $94/COMPONENT ±20%	MEDIUM SCALE INTEGRATED (MSI) CIRCUITS AT $284/COMPONENT ±4%
5000	$395,000 ± 55,300	$470,000 ± 94,000	$1,420,000 ± 56,800
4000	316,000 ± 44,200	376,000 ± 75,200	1,136,000 ± 45,400
3000	237,000 ± 33,200	282,000 ± 56,400	852,000 ± 34,100
2000	158,000 ± 22,100	188,000 ± 37,600	568,000 ± 22,700
1000	79,000 ± 11,100	94,000 ± 18,800	284,000 ± 11,400
500	39,500 ± 5,500	47,000 ± 9,400	142,000 ± 5,700
100	7,900 ± 1,100	9,400 ± 1,900	28,400 ± 1,100

It is assumed that the intent of this feasibility study is to determine whether Sneak Analysis can be modified to handle small systems effectively, at reasonable cost, and with little schedule impact. It is our considered opinion that Sneak Analysis can indeed be adapted to handle small systems. The problem is more related to the upward limits on size and mix of system components. If a system of 5000 components is not a small system, then the question can be asked as to what does constitute a small system. Project cost, manhour requirements, and schedule duration can be used to define a small Sneak Analysis application as one composed of 100-200 components based on a relay logic or generalized mix. A system of 100 components (devices) in the MSI category represents a complex and complicated configuration that although small in numerical level would still not be considered a small or inconsequential analysis effort.

One final consideration concerning the sizing level for this feasibility study is the application of automation aids in performing the Sneak Analysis task. Very small tasks can be performed effectively in a manual approach, but above the threshold limit of 100 components, establishing detailed interconnections and performing the function pathing become very difficult and time consuming. Repeatability of system configurations and confidence in the network trees decrease above this low threshold level.

3.6.2 Task Results. A determination was made during the feasibility study that two approaches could be taken to develop simplified or modified Sneak Analysis techniques for small scale systems. The approaches are:

1. Manual - for systems composed of 100 or fewer components

2. Automated - for systems composed of more than 100 components.

For each of the two approaches, the relevant tasks are defined in Section 3.4.2.2. A summary of the feasibility study by approach and task element is presented below in Table 3-55. The assumption has been made that the manual techniques and automation aids are available for use in each respective approach and that skilled analysts are performing the task. System and technique requirements are presented in the next major section.

TABLE 3-55. FEASIBILITY STUDY TASK RESULTS

APPROACH TYPE	TASKS	SIMPLIFICATION AREA	SCHEDULE IMPACT	PROGRAM COST	EFFECTIVE-NESS
	ACQUISITION	N	O	O	A
	PRE-ANALYSIS	N	O	O	A
MANUAL	PARTITIONING	N	O	O	A
	TREE DRAWING	N	-	>	L
	ANALYSIS	N	-O	>	A
	CHANGES	N	-	>	L
	ACQUISITION	Y	O	O	A
	PRE-ANALYSIS	N	O	O	A
	PARTITIONING	N	O	>	A
AUTOMATED	CODING	Y	-	>	H
	DATA PROCESSING	Y	-	>	A
	PLOTTING	Y	+	<	H
	ANALYSIS	Y	+	<	H
	CHANGES	Y	+	<	H

FOOTNOTES: Y= YES, N= NO; + - SHORTENS, O - NO CHANGE, - - LENGTHENS; > - NET INCREASE, O - NO CHANGE, < - NET DECREASE; H - HIGH, A - AVERAGE, L - LOW

3.6.2.1 Manual approach task results. For very small applications (100 components or less), the six basic tasks shown in Table 3-55 can be performed in a manual mode. All documentation will be gathered, logged and annotated as necessary to establish the design baseline to be analyzed. The overall system will be preanalyzed to determine the general or top level functions performed, the system interfaces, and any missing but required documentation. Partitioning will be performed to isolate specific and detailed functions within the system configuration drawings.

The network trees will also be cross-referenced and appropriately annotated to illustrate all cause and effect relationships and all equipment functions. Problems will be reported as identified. Change drawings will then be cataloged and the change data incorporated into the network trees. Cross-references and functional remarks will be modified and the resulting trees re-analyzed. This is the standard process in the manual approach.

1. Simplification Area - Of the six manual approach tasks shown, there are no apparent areas of simplification that can be achieved over the current approach. Even in small system applications, the hallmarks of Sneak Analysis are thoroughness, consistency and a systematic approach. To shortcut any of the steps would degrade the analysis effectiveness. One possible approach to simplifying the process would be to analyze the functional or high level composite system drawings instead of the detailed drawings that comprise the system. This is a compromise, however, to achieve cost, schedule and possible manning reductions at the expense of a thorough analysis. Several projects have been conducted in this manner at the buyer's direction. The problem with conducting the analysis at this level is that typically numerous discrepancies exist between the functional configuration and the detail schematic configuration. The number and type of discrepancies vary by project and by supplier. Consequently, the higher level drawing configuration may not adequately reflect the real configuration and adversely affect the analysis effectiveness. To achieve confidence in the functional level drawings, it should be standard practice to verify the functional level drawings to the detail drawings and identify any discrepancies. The cost to perform this comparison would virtually result in the identical cost to perform the analysis at the detail level in the first place.

2. Schedule Impact - Schedule impact for the manual approach is primarily affected by the tree drawing and change analysis tasks. Even though the system may be small, the network tree drawing phase can occupy a disproportionate share of the overall effort. The incorporation of changes is especially difficult in the manual approach. The location of specific points, wires and components in the network trees must be available on the tree or in the functional documentation to determine where to add, delete or revise the circuit configuration. As project size increases beyond the 100 component threshold, longer schedules or periods of performance are required for Sneak Analysis.

3. Program Cost - Program costs will be higher for the manual analysis approach than for the corresponding automated approach, except for the very small project applications. Increased manpower will be required to complete the tree drawing and change analysis tasks, resulting in a higher program cost. As a project size increases

significantly above the 100 component threshold level, the manpower level required for these tasks increases along with the project duration, resulting in a net increase in program cost. If the mix of components contains a high percentage of digital devices, then timing diagrams will have to be constructed which verify circuit timing. The timing diagrams are assumed to be generated manually. The generation of the timing diagrams, network trees, and the possible lack of consistency in the network trees results in the postulated increase in the analysis task cost.

4. Effectiveness - Small scale manually conducted Sneak Analysis projects are proportionately as effective as larger automated projects. The effectiveness of manually conducted analyses, however, drops off as system size increases. In addition to size, the effectiveness of the analysis is very dependent on the accuracy of the network trees. If the trees are incorrect, important problems can be missed by the analyst, probably the most severe situation discussed so far. Faulty trees can also impede efficiency and reduce contractor credibility when problems are reported that are legitimately not problems. Project time, procuring activity time, and possibly prime contractor time can be wasted evaluating reports based on incorrect trees. Individual trees can be properly analyzed but the key failing of the manual analysis approach is the inability to see all cause and effect relationships in the circuitry. This failure is based on the difficulty associated with tree cross-referencing and annotating the affected network trees. Unapparent functional relationships can occur at inconspicuous locations in circuitry and unless all cross-references are generated, some problems will be missed. Again the basic tree drawing function and modification (change analysis) are the main areas affecting effectiveness.

3.6.2.2 Automated approach task results. For applications larger than the 100 component threshold, the eight basic tasks shown in Table 3-55 should be performed with the aid of automated techniques. It will be assumed in this study that all detailed program steps will be manipulated by a user friendly interface. User friendly interface in this context indicates a software package that makes the program, the job control language, computer equipment resources, and other software related setup and execution functions transparent to the user. The user merely signs on the CRT terminal with an approved password, selects functions from a menu and perf[illegible] desired processing by a predesigned set of user prompts. The us[illegible] respond[illegible] the English type questions and all of the detailed computer program requirements are accomplished automatically. The user can concentrate on the specific processing function without need to understand program language, structure, options, resource requirements and program linkages.

All documentation can be encoded into the program system under control of a user friendly interface routine which will automatically format, edit, sort and report documentation received and used in the baseline and change analyses. Pre-analysis and partitioning are performed in virtually the same manner as in the manual approach. The coding, processing and plotting tasks generate the network trees, all cross-references and all network tree annotation. The analysis task is performed primarily in a manual manner, but some tasks including timing analyses for digital devices can be performed with linked software. Problem report generation is similar to the manual approach. Automated change analysis can be accomplished in a much more straightforward, organized and easier approach than the manual approach. Use of automated aids is the approach used in the vast majority of projects listed in the Sneak Analysis Project History Tables.

1. Simplification Area - All but two of the eight basic automated approach task areas can be modified to provide a simplified Sneak Analysis technique. The logging part of the data acquisition task can be simplified by creation of a friendly user interface routine. All documentation can be automatically recorded, the latest documentation reported and all change documentation identified and reported. The program system would also maintain a record of all documentation identified as necessary for the task but not yet received by the performing contractor. Pre-analysis and partitioning are two tasks which do not lend themselves to any type of rigid automation. The two tasks are largely engineering subjective and should remain so.

   To simplify the encoding task, the use of digitizers and component libraries should be considered. To avoid the extreme detail found at the encoding level, these features would allow the analyst to key in the device or component, the interconnecting wiring and any special features or functional remarks that describe the system component configuration. The data entry system would also be under control of a user friendly interface routine. Separate routines would be available which would check the validity of all input data, reporting all inconsistent and erroneous data. Once the input data has been checked and corrected, the mainline program processing can be invoked by user command. If the mainline computer programs are intended to be installed in a small mini- or micro computer, the source code programs would have to be modified or rewritten. The resulting program system would process the data and transfer the results to an on-line or off-line plotter device which would output hardcopy network trees. These network trees would be pathed, structured, leveled, annotated and cross-referenced. Some automated circuit analysis could be performed on the resulting network trees by inspection and identification of components and tree topology. Identification of digital devices in a tree could invoke the timing analysis routine to automatically or semi-automatically produce timing diagrams. Problem reporting could be minimally aided by merely highlighting the problem area(s) on the plotted network trees.

The change analysis process can be particularly accommodated by the automated approach. All change documentation can be rapidly entered into the logging program, the actual change information entered into code, data processing invoked, and plotter output generated for only those areas of change. A new cross-reference set is normally required when component changes occur. The subsequent analysis and reporting are the same as in the baseline analysis, except only a limited set of network trees are produced which the analyst needs to evaluate. Evaluating proposed and/or implemented changes is a very simplified task when performed with the aid of a computerized system.

2. Schedule Impact - For the entire Sneak Analysis project, there should be a positive schedule impact using the automated approach. Some elements of the overall project are unaffected by the approach and this includes data acquisition, pre-analysis and partitioning. Coding and data processing are the two tasks which have a negative impact on schedule. These are additional tasks and expenses compared to the manual approach. Plotting, analysis and change incorporation, however, are tasks which should result in a significant schedule reduction. The magnitude of the schedule reduction associated with the plotting and change incorporation more than offsets the coding and data processing tasks, thereby resulting in overall schedule reduction. Some of the analysis schedule time could conceivably be reduced by minimizing the number and type of checks based on the components present in the tree structure.

3. Program Cost - Overall cost to the program, assuming prior development and training, should be lower for the automated approach than the manual approach. In effect, the extra costs for manpower and computer processing costs are offset by the greater costs to manually construct the network trees, incorporate changes, and perform some system analysis. Many of the functions can be more completely and consistently performed in the automated approach, resulting in more attention to the analysis task and thus more return for the project dollar investment.

4. Effectiveness - Overall project effectiveness should be high, as shown in Table 3-55 for the simplified system. The automation aspects and user prompting should simplify many of the necessary but detailed project related functions so that more attention is directed toward the mainline analysis task. Recovery time associated with incorporating change data into the system should be minimized and the analyst resources directed toward evaluating the modified system configuration. The prime effectivity consideration, however, is not necessarily cost per report, reports per equipment type, or reports per program development phase. The prime consideration is the use of an automated approach to ensure a more thorough analysis and less likelihood of missing reportable conditions.

NOTE: Effectiveness of the manual and automated approaches for "small" systems can be adversely affected by the following factors:

a. The system being analyzed may be too small to identify the cause and effect relationships.

b. Interfacing equipment beyond the system bounds will have to be postulated which would disguise the real system configuration.

c. Limited clues would be applied resulting in some degradation of analysis effectiveness.

d. Training can be a decidedly negative factor because of the initial and continuing need to keep personnel abreast of current technology and the unique perspective of the analysis approach.

3.6.3 System development requirements. The system development requirements for the manual and automated approaches are presented in this section. The requirements are summary statements and would require additional expansion and detail to translate into detailed system requirements. The intention here is primarily to convey the proposed system concepts.

3.6.3.1 Manual approach requirements. In both approaches, training films and documentation would be beneficial to convey the top level and detail analysis concepts. Since the Sneak Analysis task may be performed at remote sites, the introduction and training in the project tasks would be essential to an effective analysis. The training films and documentation would have to be sufficiently detailed and complete to act in a stand-alone mode.

Detailed procedures would have to be developed which list the necessary steps to be accomplished for each task element. Adequate descriptive material would have to be supplied which would show why the step is necessary and include examples to further illustrate the concept. Because Sneak Analysis offers a unique perspective of system circuitry (and software), the documentation effort would necessarily be extensive.

The procedure development for documentation logging would be the most simple of the procedures to accomplish. Pre-analysis and partitioning procedure development are conceptually simple, but in actual performance are difficult to document. System component mix and size have definite bearings on the approach, but the configuration (hook-up) of components also influences the approach. The effort would be largely one of documenting engineering judgment.

The first of two major procedure developments involves network tree generation. These procedures would have to address component symbols, tree structure, tree leveling or orientation, function annotation, and cross-references. The mechanics of tree drawing are reasonably straightforward. Leveling of the tree to demonstrate circuit function development and propagation involves some engineering judgment. The approach would also need to demonstrate symmetry and introduce techniques to minimize crossed lines in the trees. Practice examples would have to be included which step a user through different circuit and component configurations. Examples would have to include typical power generation and distribution through switching logic, the main circuit functions involving electrical leads and switching devices in relay, digital and analog equipment, and finally the electrical return circuitry.

The second major procedure development concerns the analysis approach. The primary aspects would include pattern recognition, clue lists, function identification and propagation, cross-tree cause-and-effect relationships, and problem identification. This task would appear to be boundless due to the nature and type of problems that can be generated by a system circuit. However, the approach would have to be limited to generic conditions, with specific examples provided which illustrate the various conditions. The analyst would be cast into a highly active role, with all pattern recognition, clue application and problem identification dependent on introductory training, native intelligence, and experience. This development effort would be very involved even when a limited set of clues are considered.

3.6.3.2 Automated approach requirements. The idealized automated approach should be designed to minimize the tedium associated with the Sneak Analysis task and the requirements for detailed knowledge for each step. The approach should also be designed to minimize user time and effort in primarily the network tree generation and change modification functions so that more task emphasis can be devoted to analysis. The automated system should be self-guiding with easy to understand prompts that sequence the user through a consistent and thorough process.

1. Computer System Composition - The host computer could be installed on a large-scale computer, mini-computer, or micro-computer. The equipment selected to some extent would determine the speed of processing, the amount of data that could be processed, and the amount of sophistication that would be available through the user friendly interface. The main driving factor would be the computer programs that perform the network tree generation and plotting functions. Existing codes can be installed on mini-computers and large scale computers. Rewriting of program code would have to be performed for a micro-computer application.

The idealized computer system would be composed of the equipment shown in Figure 3-14. A digitizing device would be used strictly as a medium to enter data. The device would have defined symbols, blocks of code for each symbol, movable coordinates to layout or trace components, and linking routines that allow the user to establish the detailed wiring hookups between devices. The user would input one page of a drawing or schematic at a time by indicating devices and interconnects. The software would automatically generate the corresponding Sneak Analysis code in a form usable by the Sneak system programs.

The CRT controllers would be the primary user devices for the network tree generation task. The CRT's would be designed to allow the user to bring up processing step menus, descriptive narrative, yes-no type questions, data file contents and possibly some on-line computer plots. The user would not see detailed job control language, source computer program code, and detailed input data requirements and formats. This type interface would greatly simplify the analysis process and the analyst educational and experience background requirements. The CRT could also be used as a data input device with component libraries similar to those associated with the digitizer. All system processing would be controlled by the CRT's.

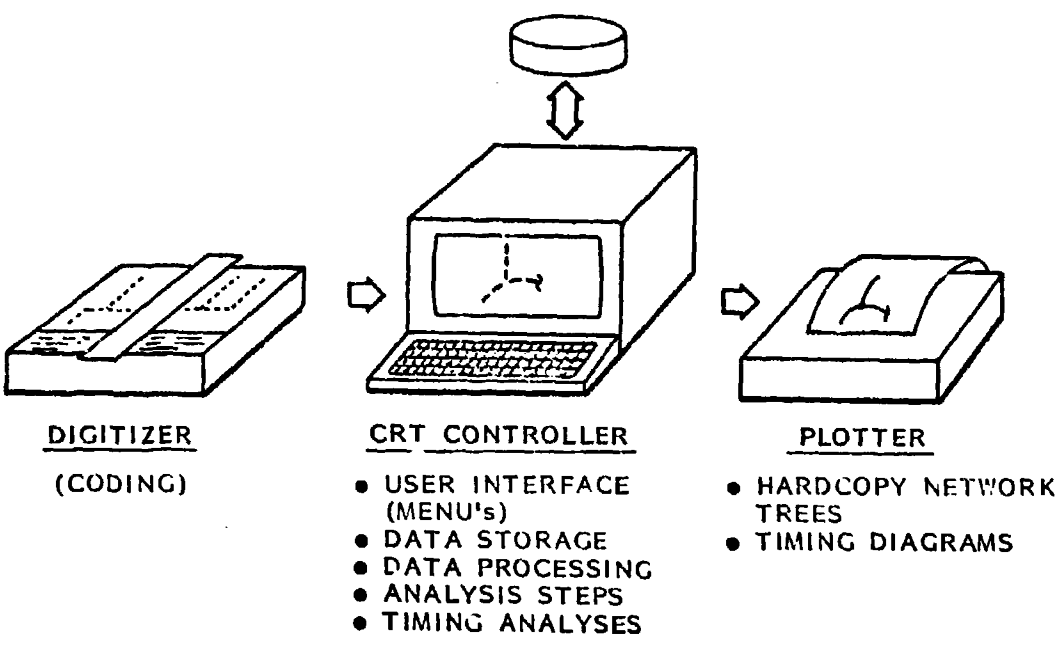


Figure 3-14. Computer System Equipment Composition

The on-line or off-line plotting devices would produce hardcopy network tree plots. Selection of these devices would be dependent on the volume and frequency of network tree production. Automated scaling of drawings is an additional important consideration because network tree size can only be controlled to a limited extent by the pre-analysis and partitioning functions.

2. Software Requirements - The software for this particular application should perform the following functions:

   a. Accept user passwords - This includes the user sign-on or log-on code and a protected macro or reserved word that would initiate the display of a Sneak Analysis processing function menu.

   b. User Friendly Interface Routines - These routines would link all program code and control the sequence of program operations. The interface would provide descriptive text and prompts to guide the user through the processing steps.

   c. Stored Component Libraries - Computer code for individual components would be stored on disc and accessed by number code from the digitizer or by a generic name or part number from the CRT console. The library would have to be updatable to incorporate new or modified devices.

   d. Core Programs - The basic Sneak Analysis code would either have to be procured or written to perform the mainline data processing. This would include all data entry, formatting, editing, error reporting, merging, pathing, sorting, report writing, plotter file generation, time analyses, pattern recognition routines, and clue list displays.

   e. Network Tree Generation and Modification - The network trees for the basic data file should be generated and plotted. The capability should exist to modify tree layout in the likely event that a different network tree perspective is required. In addition, the capability to split network trees or combine two or more trees would be necessary and useful in subsequent analysis phases.

   f. Pattern Recognition Routines - This code would inspect the resulting network trees for the presence of particular node topographs which are basic circuit patterns. Any system circuit or portion of circuitry can be decomposed into these basic patterns. The pattern recognition routine would be a driver routine along with a component identification routine that would result in specific clue list displays.

g. Network Tree Analysis - A generic clue list or questions associated with the topographic patterns would be displayed to the user to guide the analysis effort. The display would be narrative in nature and the response would not direct any additional processing.

h. Timing Analysis - A network tree reference list of all digital devices would be automatically generated and displayed to the user so that applicable areas could be interfaced to the digital logic timing analysis routine that would provide timing diagrams. Detailed libraries of system devices would have to be compiled and stored on disc for this function.

SECTION 4

4. CONCLUSIONS

Sneak Analysis is an especially effective problem identification tool which can be included in MIL-STD-785B reliability program plans to improve systems and overall program reliability. The analysis tool adds a new dimension to assessing and evaluating reliability of new or mature systems. Most of the reliability tools are fault or failure related. The fault related analyses are based either on determining eventual system effects produced by specified equipment failures, or alternately identifying undesirable top level system events and then determining those functions which can produce the undesired events. Sneak Analysis, however, provides a critical review of systems based on intended modes of operation and assumes no equipment or code failures. The identification of unintended modes of operation and their resultant system effects is the end product of a Sneak Analysis task. As such, Sneak Analysis is complementary to the fault related techniques. The analysis is best performed by a contractor, agency or group independent of the equipment or program instruction design group. Sneak Analysis should be based on actual system design produced from "as-built" drawings, schematics and wire lists for hardware systems and from "as-coded" computer program source code for software systems.

This study effort has resulted in the collection of a significant amount of information on past Sneak Analysis efforts (presented in Appendix A) which verify the original objective of the study. Sneak Analysis can identify problems before they occur in test or operation so that the cost to modify or redesign should be decreased and the reliability and safety of the system should increase. The analysis tool can be specified, program dollars allocated, and scheduled early enough in the program development cycle to allow cost-effective system changes to be implemented. One very important finding of the study effort is that regardless of the program development phase, application environment, equipment/software type, criticality ranking, or program cost, Sneak Analysis identifies a significant number of system problems. The problem report levels are typically high in early development phases and taper to lower, but significant, levels in late development phases. Man and mission critical systems represent areas of high problem report levels.

Rough order of magnitude costs for Sneak Analysis can be estimated by the procuring activity based on the system or software composition. Due to limited program dollars allocated to reliability analyses, it may be necessary to reduce Sneak Analysis project scope to selected systems or equipment functions. Tailoring of the analysis can reduce the scope of the effort and bring the cost to more acceptable levels. The cost of the analysis is high because of depth of analysis associated with the technique. Sneak Analysis is a detailed and systematic analysis, not a cursory analysis.

Guidelines for application of Sneak Analysis have been developed and presented throughout this document with the aim of informing prospective and current project procuring activities about the nature, function and roles of Sneak Analysis. The guidelines present pre-contract considerations, contracting methods, analysis scheduling, cost estimation, system applicability, expected results, and task monitoring activities. A thorough reading of the document will provide the procuring activity with the knowledge to effectively contract and manage a Sneak Analysis effort.

SECTION 5

5. RECOMMENDATIONS

Recommendation 1 - A major element missing from this effort which could be considered in measuring effectiveness is a method to track the resulting dispositions for the Sneak Analysis Reports. The information necessary would include the equipment/software/procedure changes made in response to the reports and their associated costs. It appears that some correlation could be established between the reports, phase of development, and change costs.

Unfortunately, most procuring activities have not supplied this information to The Boeing Company. In approximately 10% of our projects, we have gathered dispositions on some of the reports, but virtually no change cost figures. Typical report dispositions include:

a. Equipment/Software Change
b. Procedural Change
c. Acceptable Risk
d. Cancel

In the future, the performing Sneak Analysis contractor could complete a project table entry according to established guidelines and submit this data in the Project Final Report. The procuring activity would then have an added task to complete the remaining data categories after final report dispositions and costs are determined. The completed form would then be transmitted to RADC. In this way, the Appendix A Project History file could be maintained in an organized manner at minimal expense to RADC.

Maintaining these files should provide the following benefits:

a. An assessment of the criticality of independent analysis versus performance by the design group or contractor.

b. An evaluation of sneak finding capability of various contractors.

c. An assessment of approaches other than the systematic network tree technique to identify problems.

d. An up-to-date and complete file on Sneak Analysis that can be used as a guide for future applications.

Recommendation 2 - A task effort to develop an automated Sneak Analysis system involving related functions for small-scale hardware equipment applications should be initiated. The results of Task 6, Section 3.6, indicate the system composition and functions. The proposed computerized system would provide the user with the capability of entering discrete components and the unique configuration of interconnecting wiring. The software would generate the system configuration drawings and provide limited analysis checks. The profusion of relatively inexpensive computer systems makes this an ideal suggestion to incorporate Sneak Analysis at the detailed component level and at an early development phase. The software package would be available to the system designer and others involved in systems analysis and evaluation efforts.

Recommendation 3 - Investigate an approach to combining data required by the various analysis techniques onto a common data base. The performance of the various analyses results in a significant level of duplicate data acquisition efforts and duplicate system configuring efforts. If the common data base capability is developed, the file can be used by automated systems to perform design, reliability and safety checks with less program expense and a better consistency of results. Reliability analysis tools could then be scheduled and implemented directly from the data base.

SECTION 6

6. REFERENCES

1. 1980 DMS Market Intelligence Reports.

2. U. S. Military Aircraft Data Book, 1978.

3. U. S. Missile Data Book, 1980.

4. NAVSEA TE001-AA-GYD-010/SCA, Contracting and Management Guide for Sneak Circuit Analysis (SCA), Naval Sea Systems Command.

5. Glen D. Bergland and Ronald D. Gordon, Tutorial - Software Design Strategies, IEEE EHD 149-5, 1974.

6. July 23, 1981, Electronic Design Magazine, Page 75.

7. Glass, Robert L., Software Reliability Guidebook, Pages 86-104, Prentice-Hall, 1979.

8. Spectrum of Budgets/Costs for Software Life Cycle Costs, April 1981.

9. Godoy, S. G. and Engles, G. J., Sneak Circuit and Software Sneak Analysis, Journal of Aircraft, Vol. 15, No. 8, August 1978, Pages 509-513.

10. Buratti, D. L., Pinkston, W. E., Simkins, R. O., Sneak Software Analysis, 4th Digital Avionics Systems Conference, November 1981, Pages 206-211.

APPENDIX A

SNEAK ANALYSIS PROJECT HISTORY TABLES

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY SPACE

PAGE # 1

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
APOLLO/ASTP (Multiple Vehicles)	$20 - BILLION	Electrical Power Distribution Controls, Electrical Support Equipment Interface	I/C^3	Relay	Full Scale Engineering Development FSED	Hardware, Procedures, Mission Support, Fluidics Changes	2,413	208 SCR's 13 DCR's 1500 DER's	1967-1975 95 Months
SKYLAB	1.5 BILLION	Electrical Power and Control, Experiment Package, Electrical Support Equipment Interface	I/C^3	Relay	FSED	Hardware, Grounding Procedures, Mission Support, Changes	1,993	259 SCR's 307 DER's	1971-1974 37 Months
SATURN 1-C	$920 MILLION	Electrical Power and Control/Electrical Support Equipment Interface	I/C1, C2	Relay	Unlimited Production	Hardware	200	EPS 7 DCP's ESE 20 DCR's	1971-1972 18 Months
BURNER II	$19 MILLION	Flight Sequencer	II/C2	Relay	FSED	Hardware	5	2 SCR's 4 DCR's 6 DER's	1972 7 Months
ATS-F	$375M/ $16M	Electrical Power, Sequencer, Experiments, Telemetry, Propulsion, Controls	II/C^3	Relay, Digital	FSPD	Hardware, Changes, Procedures	~133	55 SCR's 67 DER's	1972-1974 29 Months
SPACE SHUTTLE (Multiple Vehicles)	$6 - BILLION	ALL subsystems, Electrical Support Equipment Interface	I/C^3	Relay, Digital, Analog, Microprocessor	Full Scale Engineering Development	Hardware, Changes	5,425	124 SCR's 85 DCR's 728 DER's	1974-Present 86 Months
VIKING LANDER	$950M	Power Control and Distribution, Data Acquisition and Processor Unit	II/C1, C2	Relay	Unlimited Production	Hardware, Procedures	35	14 SCR's 23 DER's	1974 4 Months
DELTA LAUNCH VEHICLE	$1.4 B	Electrical Power Distribution, Electrical Support Equipment Interface	II/C2	Relay	Unlimited Production	Hardware	63	16 SCR's 10 DCR's	1974-1975 7 Months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY SPACE

PAGE # 2

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
GLOBAL POSITION-ING SATELLITE	$350M	Global Positioning System and Ground Test Equipment	II/C2, C3	Relay, Digital, Analog	Pilot Production	Hardware, Changes	257	58 SCR's 44 DCR's 134 DER's	1975-1977 15 Months
NTS-2 SATELLITE	$350M	Excluded Telemetry, Navigation and Power Generation	II/C2	Relay, Digital	Pilot Production	Hardware	11	6 SCR's 13 DCR's 62 DER's	1976 4 Months
DEFENSE METEORO-LOGICAL SATEL-LITE	$30M	Control Interface Unit, Launch Sequencer	II/C2	Digital	Pilot Production	Hardware	131	12 SCR's 19 DCR's 27 DER's	1976 4 Months
LGM-30G MISSILE	$15 - BILLION	Monitor, Control and Ordnance	I/C2	Digital, Analog	Unlimited Pro-duction	Hardware, Accident Analysis, Power & Load Analysis	130	1 SCR 4 DCR's 12 DER's	1976 17 months
TIROS - N/502	$500M	Instrument, Power, Pyro-technic, Attitude Con-trol, Thermal Control	II/C2	Digital	Unlimited Pro-duction	Hardware	155	17 SCR's 7 DCR's 48 DER's	1976-1977 14 months
Minuteman III	$15 - BILLION	NS20-GYRO Torquing Loop - Portion of Guidance Electronics	I/C2	Digital, Analog	Unlimited Pro-duction	Hardware	19	1 DCR 5 DER's	1978 3 Months
MULTI-MISSION SPACECRAFT	$60M	Signal Conditioning and Control Unit	II/C3	Relay, Digital, Analog	Pilot Production	Hardware FMEA FTA CCFA	25	4 SCR's 10 DCR's 15 DER's	1978-1979 10 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY SPACE

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
LANDSAT-D	$20 MILLION *	Signal Conditioning, and Control Unit, Remote Interface Unit (RIU)	II/C3	Relay, Digital, Analog	FSED	Hardware (RIU), Review of FMEA results for Signal Conditioner	25	1 DCR 28 DER's	1979 4 Months
VOYAGER	$350M	Interfaces to RF System, Flight Data and Photopolarimeter Systems	II/C2, C3	Relay, Digital	Unlimited Production	Hardware Interfaces	28	4 DCR's 11 DER's	1979 4 months
DESIGNATED OPTICAL TRACKER (DOT)	$4.3M	Interface Unit	II/C2	Digital	FSED	Hardware	13	1 SCR 5 DCR's 14 DER's	1976-1977 2 Months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE # 1

PROJECT	PROGRAM CONTRACT VALUE	COMPONENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
COMPASS COPE	$156M	Electrical Power Control and Distribution	II/C^{3}	Relay	FSED	Hardware	14	39 SCR's 29 DCR's 32 DER's	1973 6 Months 1974 6 Months
347 HEAVY LIFT HELICOPTER	$13M	Fly-By-Wire Flight Controller	I/C1, C2	Relay	Unlimited Production	FTA, Hardware	21	1 SCR	1973 3 months
737 A/C	$1.5B	Research Support Flight System	I/C1, C2	Relay	Unlimited Production	Hardware	15	24 SCR's 6 DCR's 14 DER's	1974 3 months
F-4C	$20B	Flight Control System	I/C1, C2	Relay	Unlimited Production	Hardware Component Failure	36	4 SCR's 14 DCR's 14 DER's	1974 4 months
747	$7B	Landing Control and Logic Unit	I/C2	Digital	Unlimited Production	Hardware	11	3 SCR's 4 DCR's 33 DER's	1974 1 month
AWACS	$2.5B	Electrical Power System	I/C2	Relay	FSPD	Hardware	22	57 SCR's 2 DCR's 19 DER's	1974 3 months
AQM-34M DRONE	$20M *	Electrical Power Distribution	II/C^{3}	Relay	FSED	Hardware	37	6 SCR's 9 DCR's 19 DER's	1974 - 1975 5 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE # 2

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
SHUTTLE TRAINING AIRCRAFT	$25M	Modified and New Electrical Power System	I/C2	Digital	FSED	Hardware	98	40 SCR's 46 DCR's 181 DER's	1975 7 months
F-8 A/C FLY-BY-WIRE	$50M *	Wiring, Instrumentation Power, Primary & Backup Computer	I/C1, C2	Relay, Digital, Analog	FSED	Hardware	39	19 SCR's 7 DCR's 28 DER's	1975-1976 12 months
FMU-112B FUZE	$50M	Fuze and Electrical System	II/C2	Digital	FSED	Hardware	15	3 SCR's 10 DCR's 4 DER's	1975 4 months
B-52G/FB-111A/ WEAPON CIRCUITS	$6B	Aircraft Monitor and Control Interface with Nuclear Weapons	I/C2	Relay	Unlimited Production	Hardware, Power and Load Analysis	140	5 SCR's 2 DCR's 4 DER's	1975 4 months
ALQ-99	$94M	Electronic Countermeasure System - Power, Encoder, Receiver	III/C2	Digital	FSED	Hardware	28	25 SCR's 7 DCR's 45 DER's	1975-1976 4 months
ALCM	$60M	Seco[illegible]ry Power, Ord[illegible]e, Flight Cont[illegible], Instrumentation	II/C2	Relay	FSED	Hardware	40	7 SCR's 3 DCR's 16 DER's	1975-1976 6 months
F-4/APQ-120	$20B	Fire Control Radar System	III/C2	Digital	Unlimited Production	Hardware	105	6 SCR's 12 DCR's 19 DER's	1976 8 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
F-4E	$20B	Monitor, Control and Release System	II/C2	Relay	Unlimited Production	Hardware Power and Load Analysis	140	1 SCR 1 DCR 11 DER's	1976 7 months
TERMINAL CONFIGURED VEHICLE	$106M	Elevator and Rudder Control System in Pitch Axis Program	I/C1 C2	Assembly	Pilot Production	Software, Study Effort	45	20 SSR's 1 SDCR 3 SDER's	1976 9 months
VTOL APPROACH AND LANDING TECHNOLOGY HELICOPTER (CH-47) (VALT)	$15M *	Control Transfer System	II/C2	Relay	Unlimited Production	Hardware	11	4 SCR's 2 DCR's 2 DER's	1976 5 months
F-4 ASN-63	$20B	Portions of Inertial Navigation System	III/C2	Digital	Unlimited Production	Hardware Failure Rate Component Analysis	94	6 SCR's 4 DCR's 19 DER's	1976 6 months
AIRBORNE LASER LAB	$200M	Cycle III Electrical System	II/C3	Relay	FSPD	Hardware Component Failure Analysis	148	18 SCR's 12 DCR's 21 DER's	1976-1977 15 months
F-18 RADAR	$7B	Power Supply, Distribution; Transmitter, Antenna	III/C2	Digital	FSED	Hardware	75	24 SCR's 11 DCR's 31 DER's	1976-1977 8 months
ARC-164	$300M	UHF Radio	III/C1 C2	Digital	FSPD	Hardware	66	1 SCR 3 DCR's 8 DER's	1977 5 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE 8 4

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
GBU-15	$75M	Weapon Analog Auto-pilot and Demand Actuator Control Unit	III/C3	Digital, Analog	FSPD	Hardware	40	6 SCR's 10 DCR's 1 DER	1977 5 months
GBU-15	$75M	Weapon Control Unit	I/C3	Digital, Microprocessor, Assembly	FSPD	Hardware/Software	62 (40/22)	HW 17 SCR's 17 DCR's 28 DER's SW 3 SSR's 8 SDCR's 20 SDER's	1977 5 months
F-4E	$20B	Angle of Attack Transmitter	III/C1,C2	Analog	Unlimited Production	Hardware	19	5 SCR's 6 DCR's 4 DER's	1977 8 months
F-16	$13B	Fire Control Radar	III,C2	Digital, Microprocessor	FSPD	Hardware	99	17 SCR's 34 DCR's 65 DER's	1977-1978 6 months
XM-130	$14M	CHAFF/FLARE Dispenser System	III/C1,C2	Analog	Pilot Production	Hardware	30	3 SCR's 6 DCR's 5 DER's	1977 3 months
F-14	$10B	Automatic Flight Control System	III/C1,C2	Relay	Unlimited Production	Hardware	119	3 SCR's 6 DCR's 37 DER's	1977-1978 11 months
A-7D	$2.3B	DIGITAC I Multimode Flight Control System	III/C1,C2	Relay, Assembly	Pilot Production	Hardware/Software Interface	~ 96 (68/28)	HW 5 SCR's 6 DCR's 6 DER's SW 10 SSR's 10 SDCR's 4 SDER's	1977-1978 9 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE # 5

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
F-4C	$20B	Nose Wheel Steering	I/C1, C2	Relay, Analog	Unlimited Production	Hardware FTA FHA	20	2 SCR's 7 DCR's 1 DER	1977 2 months
F-104G	$15M *	DCU-201/GMM-4 Nuclear Weapon Controller/Tester Sys.	II,C2	Relay, Digital	Unlimited Production	Hardware	40	4 SCR's 2 DCR's 2 DER's	1977 3 months
SHRIKE AGM-45-10	$500M	Electrical, Guidance, Armament, Control, Interface to F-4E Launcher	II/C2	Relay, Digital	Unlimited Production	Hardware	44	2 SCR's 4 DCR's 9 DER's	1977-1978 6 months
ALCM	$200M	Radar Altimeter Element	III/C2	Digital, Analog	FSPD	Hardware	42	4 SCR's 18 DCR's 54 DER's	1977-1978 5 months
F-14	$10B	Electrical Power Distribution	I/C2	Relay	Unlimited Production	Hardware	127	3 SCR's 4 DCR's 26 DER's	1977-1978 8 months
AN/ALQ-156	$20M	Missile Detector System	III/C2	Assembly	FSED	Software	11	2 SSR's 8 SDCR's 2 SDER's	1978 4 months
F-16	$13B	Flight Control System/ Electrical Power Distribution	I/C1, C2	Relay, Digital, Analog	FSPD	Hardware	256	2 SCR's 43 DCR's 119 DER's	1978 8 months
HELLFIRE LAUNCHER	$160M	Launcher Electronic Command Signal Programmer	II/C2	Digital, Analog	Pilot Production	Hardware	72	1 SCR 3 DCR's	1979-1980 8 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE # 6

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
T-90	$5M *	DC Generator System	I/C2	Relay, Digital	FSPD	Hardware	13	5 SCR's 13 DCR's 28 DER's	1978 1 month
SM-2	$1B	Guidance Computer System	II/C2	Relay, Digital, Analog, Micro-processor, Assembly	Unlimited Production	Hardware/Software Interface	111 (70/41)	HW 5 SCR's 22 DCR's 45 DER's SW 5 SSR's 10 SDCR's 2 SDER's	1978 5 months
F-111F	$6B	DCU-218 Controller/ Weapon System	II/C2	Relay, Digital	Unlimited Production	Hardware	95	9 SCR's 5 DCR's 22 DER's	1978 6 months
F-100 ENGINE	$15B	Digital Electronic Engine Control System	II/C2	Digital	FSPD	Hardware	51	7 DCR's 23 DER's	1978 5 months
HELLFIRE TELEMETRY MISSILE	$95M	Autopilot Laser Seeker Control	II/C2	Digital, Analog	FSPD	Hardware	83	3 SCR's 12 DCR's 28 DER's	1978-1979 7 months
FALCON JET	$205M	Electrical Power Distribution System	I/C2	Relay, Analog	FSPD	Hardware FMEA, FTA, CCFA	38	2 SCR's 7 DCR's 7 DER's	1979 6 months
LAMPS	$1.6B	Avionics System	II/C2	Digital, Analog	Pilot Production	Hardware	300	22 SCR's 11 DCR's 48 DER's	1978-1979 15 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
GULFSTREAM II	$25M	Flight Control, Thrust Reverser, Fuel Control Systems	III/ C1,C2	Relay, Digital, Analog	Pilot Production	Hardware Changes FMEA, FTA, CCFA (Primarily Change Analysis)	40	1 DCR	1978-1979 9 months
F-16	$15B	Weapons Control Mechanization System	II/C1, C2	Digital, Microprocessor, Assembly	Unlimited Production	Hardware/Software Interface, Power and Load Analysis	461 206/255	HW 10 SCR's 39 DCR's 161 DER's SW 9 SSR's 13 SDCR's 46 SDER's	1979-1980 13 months
AIM/RIM-7M (SPARROW)	$1B	Portions of Guidance and Control	II/C2	Digital, Analog	FSPD	Hardware	127	1 SCR 14 DCR's 18 DER's	1979 6 months
AGM-88A HIGH SPEED ANTI-RADIATION MISSILE	$175M	Power and Control	II/C2	Relay, Digital, Analog	FSPD	Hardware	92	3 SCR's 6 DCR's 15 DER's	1979 5 months
F-18	$15B	Flight Control Electronics System/ Electrical Power Dist.	I/C2	Digital, Analog, Microprocessor	FSPD	Hardware	426	FCS 1 SCR 9 DCR's 22 DER's EPS 10 DCR's 2 DER's	1979-1980 11 months
F-15, F-16, F-18, A-7	F-15 $10B F-16 $15B	ACES II Ejection Seat Recovery Sequencer	I/C2	Analog	F-16 Pilot Production F-18 FSPD F-15 Unlimited Production	Hardware	10	1 SCR 3 DCR's 1 DER	1979 4 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY AIRBORNE

PAGE # 8

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
737/727	$15B *	Auto Performance Reserve System (Microprocessor)	III/C2	Digital, Microprocessor, Assembly	FSPD	Hardware/Software Interface	15 (5/10)	HW 5 SCR's, 4 DCR's, 17 DER's; SW 3 SSR's, 10 SDCR's, 6 SDER's	1979; 3 months
F-100 ENGINE	$15B	Digital Electronic Engine Control System	III/C2	Assembly	Pilot Production	Software	76	15 SDCR's, 7 SDER's	1979, 4 months
AWACS	$3B	Flight Engineer Panel Lighting System	III/C1	Relay	Unlimited Production	Hardware	5	1 SCR, 2 DCR's, 16 DER's	1979; 1 month
ROTOR SYSTEMS RESEARCH AIRCRAFT	$25M *	Electronic Flight Control System	II/C1, C2	Digital	FSPD	Hardware, FMEA, FTA, CCFA, System Safety Handbook	97	5 SCR's, 1 DCR, 13 DER's	1979-1980; 16 months
HELLFIRE TACTICAL PROTOTYPE	$220M	Autopilot Electronics	II/C2	Digital, Analog	FSPD	Hardware	45	1 SCR, 5 DCR's, 11 DER's	1980; 5 months
SHUTTLE TRAINING AIRCRAFT	$25M	Flight Control Computer	II/C2	Assembly	Pilot Production	Software	48	(In-Process) 37 SSR's, 55 SDCR's, 115 SDER's	1980-1981, 7 months
PROTOTYPE MINIATURE AIR-LAUNCHED SYSTEM	$200M	Interface Deployment, Actuation, Firing and Range Safety for upper stage and miniature vehicle	II/C2	Relay, Digital, Analog	FSPD	Hardware	60	(In-Process) 6 SCR's, 6 DCR's, 6 DER's	1980-1981, 9 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY GROUND/WATER

PAGE # 1

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
LOCAL FREQUENCY CONVERTER	$2M	Power Line Frequency Converter	III/C2	Relay	Unlimited Production	Hardware	3	3 SCR's	1972 3 months
SAVANNAH RIVER NUCLEAR PLANT	$200M	Automatic Incident Action System Safety Circuits	II/C1, C2	Relay	Unlimited Production	Hardware, Procedures	19	7 SCR's 5 DCR's 22 DER's	1973 1 month
		M-2 Containment Console						18 SCR's 13 DCR's 22 DER's	1974 1 month
MORGANTOWN PERSONAL RAPID TRANSIT I	$60M	Engineering Station	I/C3	Digital	FSPD	Hardware, Grounding	23	14 SCR's 13 DCR's 18 DLR's	1973 3 months
MORGANTOWN PERSONAL RAPID TRANSIT IB	$60M	Vehicle Control and Communication System	I/C3	Digital	Unlimited Production	Hardware Change Analysis for Engineering Station	20	10 SCR's 2 DCR's 8 DER's	1974 3 months
PERSHING I, Ia	$670M	All Block VI except Computer and Guidance, Block VII	II/C2	Relay	FSED	Hardware	160	21 SCR's 8 DCR's 63 DER's	1973-1974 14 months
HANFORD N-REACTOR	$70M	Safety and Rod Control	II/C1, C2	Relay	Unlimited Production	Hardware	45	11 SCR's 8 DCR's 13 DER's	1974 11 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY GROUND/WATER

PAGE 1 2

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
OIL WELL DRILLING SYSTEM	$7M *	Control and Diagnostic	III/C1, C2	Digital, Analog	FSED	Hardware Mean-Time-Between Failure Analysis	25	1 SCR 15 DCR's 11 DER's	1974-1975 6 months
THISTLE FIELD DRILLING PLATFORM	$2B *	Upending and Replacement Systems	III/C2	Relay	FSED	Hardware, FMEA	40	7 SCR's 4 DCR's 14 DER's	1975-1976 6 months
FAST FLUX TEST FACILITY (PLANT PROTECTION SYSTEM)	$500M *	Primary Shutdown Portion of Plant Protection System	II/C1, C2	Relay	Unlimited Production	Hardware	120	5 SCR's 13 DCR's 10 DER's	1976 7 months
AN/SQS-26 SONAR (LOUIS ALLIS POWER SUPPLY)	$97.5M	Transmitter Power Supply	III/C2	Relay	Unlimited Production	Hardware	20	4 SCR's 16 DCR's 4 DER's	1976 6 months
BAY AREA RAPID TRANSIT	$1.6B	Door Control System	I/C^3	Relay, Analog	Unlimited Production	Fault Tree, CCFA, Hardware, MTBH, Grounding	31	25 SCR's 21 DCR's 24 DER's	1976-1977 5 months
COGNAC	$2B *	Lowering System Instrumentation	II/C1, C2	Relay	Pilot Production	Hardware, FMEA	20	5 SCR's 5 DCR's 2 DER's	1977 11 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY GROUND/WATER

PAGE # 3

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $ (K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
SAVANNAH RIVER PLANT	$200M	Control and Scram Systems	I/C^3	Relay	Unlimited Production	Hardware	29	7 SCR's 5 DCR's 22 DER's	1977 5 months
INCIPIENT FAILURE DETECTION	$225K	PM-109 Software	II/C^3	Assembly	FSED	Software	3	4 SSR's 1 SDCR 1 SDFR	1977 2 months
AN/TPQ-37	$74M	Transmitter Control Unit	II/C1, C2	Digital, Analog	Pilot Production	Hardware	67	2 SCR's 2 DCR's 8 DER's	1977-1978 6 months
UNDERWATER FIRING DEVICE	$500K	Electronics	II/C2	Digital	FSED	Hardware FMEA	5	1 SCR 3 DCR's 2 DER's	1978 1 month
BAY AREA RAPID TRANSIT SCA	$1.6B	Door Control System	II/C^3	Relay, Analog	Validation	Evaluation of Hardware Change and FMEA, FTA, CCFA	72	Evaluation of Changes - No Reports -0-	1977-1978 13 months
DISCOVERER 7 SEAS DRILLSHIP	$1B	Electrical System	II/C^3	Relay	Unlimited Production	Hardware, PHA, FTA, FMEA, CCFA, Procedure	30	3 SCR's 1 DCR	1978 3 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY GROUND/WATER

PAGE # 4

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
GBU-15	$75M	Shop Maintenance Test Set	III/C1, C2	Relay, Digital, Analog, Assembly, High Order Language	FSED	Hardware/Software Interface	266 (66/200)	HW 6 SCR's, 79 DCR's, 44 DER's; SW 9 SSR's, 32 SDCR's, 1 SDER	1978-1979, 9 months
SUB-LAUNCHED MOBILE MINE	$60M	All Electronics	II/C2	Relay, Digital	FSED	Hardware	53	3 SCR's, 19 DCR's, 10 DER's	1979, 5 months
BART	$1.6B	Manual Cab Signalling System	I/C^3	Digital	FSED	FMEA, MTBH (primarily FMEA)	33	8 DER's	1979-1980, 8 months
FAST FLUX TEST FACILITY	$500M	Electrical Power Systems	II,C1, C2	Relay	Unlimited Production	Hardware Load Switching	367	16 SCR's, 4 DCR's, 134 DER's	1979-1980, 10 months
SHUTTLE	$8B	Remote Manipulator System, Master Events Controller	I/C^3	Relay, Digital, Analog	FSED	Hardware, FMEA, FTA, CCFA	6	6 SCR's, 13 DCR's, 5 DER's	1980, 2 months
OMD DATA LINK CONTROLLER	$200M *	Data Link Controller Software between Nuclear Plant Microprocessors	II/C1, C2	Assembly	FSED	Software	12	2 SSR's, 6 SDCR's, 8 SDER's	1980, 3 months
NUCLEAR PLANT PROTECTION	$200M *	Reactor Protection System (RPS-II)	II/C1, C1	Assembly	FSED	Software	49	11 SDCR's, 9 SDER's	1980-1981, 6 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY GROUND/WATER

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $(K)	REPORTS	DATES AND PERIOD OF PERFORMANCE
VERTICAL LAUNCH SYSTEM	$120M	Launch Sequencer, Motor Control System, Electrical Power	II/C2	Analog	FSED	Hardware	391	21 SCR's 21 DCR's 119 DER's	1980-1981 7 months
LAUNCH PROCESSING SYSTEM	$500M *	Portions of the Check-out, Control and Monitor Subsystem	I/C1, C2	HOL, Assembly	Pilot Production	Software	300	36 SSR's 50 SDCR's 88 SDER's	1980-1981 7 months
VOLUMETRIC INFUSION PUMP	$20M *	Complete System	I/C2	Digital, Microprocessor, Assembly	Pilot Production	Hardware/ Software	40 (20/20)	HW 5 SCR's 16 DCR's 7 DER's SW 6 SSR's 9 SDCR's 7 SDER's	1980-1981 6 months

SNEAK CIRCUIT ANALYSIS APPLICATION GUIDELINES PROJECT HISTORY

EQUIPMENT CATEGORY EXCLUSIONS

PAGE # 1

PROJECT	PROGRAM CONTRACT VALUE	EQUIPMENT/ SUBSYSTEM REQUIREMENTS	EQUIPMENT CRITICALITY	EQUIPMENT TYPE	DEVELOPMENT PHASE	TYPE(S) OF ANALYSIS	SNEAK ANALYSIS CONTRACT $	REPORTS	DATES AND PERIOD OF PERFORMANCE
AEM - SAGE						Hardware			
ADVANCED ENGINEERING TEST FACILITY						Hardware			
AIM-9L SIDEWINDER						Hardware			
COMPONENT PREPARATION LABORATORY						Hardware			
COMPONENT TEST FACILITY						Hardware			
CAPTOR						Hardware			
TITAN II (2)						Hardware			
BI-AUDIT						Hardware			

APPENDIX B

SNEAK ANALYSIS COST ESTIMATION

SECTION 1

1. INTRODUCTION

This section of Appendix B presents information to assist in developing rough order of magnitude cost estimates for the performance of hardware and software Sneak Analysis tasks. The cost estimation approach is derived from the project histories presented in Appendix A of this document. All cost figures have a 1979 cost base. This cost estimating approach is to be used for Sneak Analysis tasks that are performed at the detailed component/instruction level and use the network tree path analysis approach and certain additional proprietary enhancements described in Sections 3.3.1.2 and 3.3.3.1. The Section 2 cost estimating approach is to be used for Sneak Analysis tasks performed at the system or subsystem level and use a technique other than the network tree analysis approach.

1.1 Hardware Cost Estimating Process. The estimating process for hardware Sneak Analysis tasks is normally conducted in two phases. In the first phase, the initial scope of work is presented and example documentation is inspected to determine the approximate size of the system to be analyzed and cost of the analysis. In the second phase, the complete system of documentation is inspected and a detailed cost estimate is generated based on actual component or instruction count. Additional task tailoring can be performed in the second phase by excluding specific functions in the component or instruction count, thereby lowering costs.

1.1.1 First phase hardware estimate. The procuring activity should prepare a rough order of magnitude cost estimate based on the composition of the system, the type and amount of hardware involved, and the tentative schedule desired. Example documentation should be available for basing the estimate. The ROM will establish the general cost to perform the analysis. Based on this approximate parts count, the procuring activity can determine cost by use of Figure B-1. The cost/parts curve is based on a generalized mix of hardware components, when actual system composition is not known. Some applications primarily composed of manually switched systems would encounter lower costs than shown in Figure B-1, while highly digital systems would encounter higher costs.

1.1.2 Second phase hardware estimate. The second phase will be a detailed examination of documentation to establish a better estimate of the type and number of electrical system components. The source, type and means of acquiring all documentation for the analysis will be determined, as well as the use of any computerized systems to assist the analysis process. The generation of the detailed analysis cost will also reflect the added costs relevant to the handling of classified data and subsequent change analysis for documentation over and above the baseline system. The second phase estimate should contain an itemized listing of task elements. The cost tables are reasonably regular and support a linear cost relationship with parts count, except near the origin. Because of the differences in the amount of labor, computer time, and materials required to analyze different hardware part types, each part type has a different weighting

factor in determining the cost of the Sneak Analysis task. The cost (in 1979 dollars) can be calculated by adding together the costs for each individual part type. Table B-1 presents the weighting factors for different hardware part types and their approximate tolerances. Table B-2 presents a sample calculation for a system consisting of 1000 parts of the indicated mix ratio.

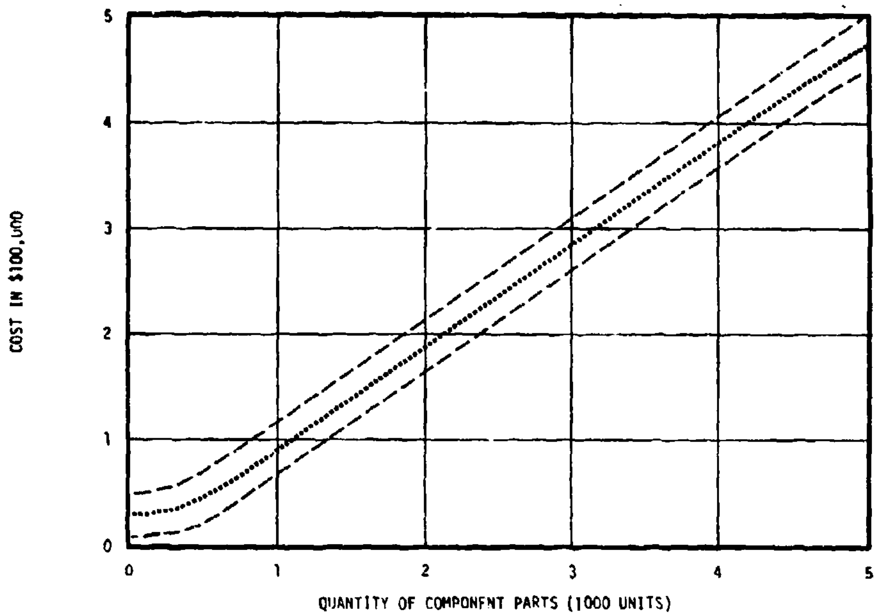


Figure B-1. SCA Cost Vs. Job Size

1.1.3 Cost adjustments. Costs calculated for Sneak Analysis tasks are stated in 1979 dollars for work generally performed in the Houston, Texas area. Cost adjustments for inflation in later years and for different geographical areas can be made using current statistics provided by the U. S. Department of Labor, Bureau of Labor Statistics (BLS). Examples of the type of data available in these publications are shown in Table B-3. These data are not necessarily current; the latest available issues of the BLS data should be consulted.

1.1.4 Hardware cost estimating accuracy. Historically, the accuracy of the parts-count technique presented in Table B-2 is ±10%. When the exact component mix is not known and the weighting factor for a generalized component mix in Table B-1 is used, the accuracy is ±20%. Both of these estimators produce larger errors for parts-count below about 300 parts. In this region, the data are better represented by a constant dollar figure of $30,000 ± $20,000.

TABLE B-1. COST FACTORS FOR DIFFERENT PART TYPES

Part Type	Weighting Factor	Weighting Factor Tolerance
	$/Part	$/Part
Resistors, Capacitors, Coils	29	±8
Relays, Transistors, Switches	79	±11
Small-Scale Integrated Circuits (SSI)	164	±14
Medium-Scale Integrated Circuits (MSI)	284	±14
Large-Scale Integrated Circuits (LSI)	468	±25
Generalized Component Mix (Used when actual component mix is not known)	94	±19

TABLE B-2. SAMPLE CALCULATIONS

Part Type	Number of Parts	X	Weighting Factor	=	Component Cost
Resistors, Capacitors, Coils	400	X	29/Part	=	$ 11,600
Relays, Transistors, Switches	200	X	79/Part	=	15,800
SSI	150	X	164/Part	=	24,600
MSI	100	X	284/Part	=	28,400
LSI	50	X	468/Part	=	23,400
Totals	1,000				$103,800

TABLE B-3. EXCERPTS FROM BUREAU OF LABOR STATISTICS PUBLICATIONS

Occupation and Level	Monthly Salaries				Annual Salaries			
	Mean	Median	Middle Range		Mean	Median	Middle Range	
			First Quartile	Third Quartile			First Quartile	Third Quartile
Engineers								
Engineers I	1,327	1,316	1,225	1,425	15,926	15,794	14,700	17,097
Engineers II	1,464	1,450	1,348	1,580	17,567	17,400	16,170	18,960
Engineers III	1,683	1,665	1,515	1,835	20,194	19,980	18,183	22,020
Engineers IV	1,998	1,983	1,800	2,185	23,972	23,790	21,600	26,220
Engineers V	2,333	2,315	2,121	2,535	28,001	27,780	25,452	30,420
Engineers VI	2,689	2,657	2,433	2,911	32,264	31,887	29,198	34,934
Engineers VII	3,043	3,005	2,774	3,291	36,520	36,056	33,288	39,492
Engineers VIII	3,509	3,450	3,172	3,758	42,104	41,400	38,062	45,101
Technical Support								
Key Entry Operators I	712	666	585	784	8,546	7,994	7,020	9,411
Key Entry Operators II	842	810	709	937	10,099	9,720	8,509	11,241

Engineer I
General characteristics. This is the entry level of professional work requiring a bachelor's degree in engineering and no experience, or the equivalent of a degree in appropriate education and experience. Performs assignments designed to develop professional work knowledges and abilities. May also receive formal classroom training.

Engineer V
General characteristics. Applies intensive and diversified knowledge of engineering principles and practices in broad areas of assignments and related fields. Makes decisions independently on engineering problems and methods. Uses advanced techniques and modifications and extensions of theories, precepts, and practices of own field and related sciences and disciplines

Engineer III
General characteristics. Independently evaluates, selects, and applies standard engineering techniques, procedures, and criteria, using judgment in making minor adaptions and modifications. Assignments have clear and specified objectives and require the investigation of a limited number of variables.

Engineer VII
General characteristics. Makes decisions that are recognized as authoritative and have an important impact on extensive engineering activities. Will have demonstrated creativity, foresight, and mature engineering judgment in anticipating and solving unprecedented engineering problems, determining program objectives and requirements, organizing programs and projects and developing standards and guides.

1.2 Software Cost Estimating Process. The estimating process for software Sneak Analysis tasks is based on the number of executable lines of assembly language instructions. The cost is approximately $10 per assembly language instruction. Software programs are typically composed of executable and non-executable code. Only the executable code is to be counted for estimate purposes.

High order languages present a problem in the estimation process. A high order language instruction is a generic or "English type" instruction that represents one or more equivalent assembly language instructions. Each language has a unique expansion factor for conversion to an equivalent assembly language basis. Very little historical data is available except for three high order languages. Additional information for estimating high order language analysis applications should be available in the near future.

The accuracy for estimating software Sneak Analysis costs using this approach is ±10%.

SECTION 2

2. INTRODUCTION

This section of Appendix B presents information to assist in developing rough order of magnitude cost estimates for hardware and software Sneak Analysis tasks using an analysis approach different than that specified in Sections 3.3.1.2 and 3.3.3.1.

2.1 <u>Cost Estimates for Other Sneak Analysis Procedures</u>. Estimating the cost of a Sneak Analysis task when new or innovative procedures are to be performed or when the scope of the task has been limited by some tailoring process is more difficult. If the technical monitor is sufficiently knowledgeable of the analysis procedure which is to be used, an estimate of cost can be derived. The cost estimate is developed by isolating each task to be performed. Preparing a Work Breakdown Structure (WBS) of the required tasks is a very useful first step. The WBS elements involved are Project Management, Data Management, Engineering Analysis, Quality Assurance, and Reporting. The procuring activity would estimate the engineering and support time involved in each WBS element, any computer charges involved, special materials, equipment charges, and travel. It is not the intent herein to provide a "cookbook" for this estimating process, but rather to identify some of the factors that should be considered.

2.1.1 Engineering skill levels required. The performance of Sneak Analysis requires an analyst possessing certain learned skills if it is to be performed efficiently. It also requires a depth of experience in electrical equipment design (or in software coding practices) which is not generally available in entry-level personnel. Most detailed electrical Sneak Analysis will be done by engineers in categories II, III, and IV as defined by the U. S. Department of Labor, Bureau of Labor Statistics. The exact mix will be dependent both on job requirements and on the engineering mix that the contractor has available at a given time. The contractor may, for example, substitute a higher engineering category for a lower one if there is an insufficient number of personnel in the lower category on his staff. Equivalent statements can be made for software analysts; personnel capable of doing software Sneak Analysis normally have titles such as "Systems Analysts" or "Senior Systems Analysts." The Department of Labor Statistics has not defined skill categories in this technical discipline.

2.1.2 Engineering time. Although Sneak Analysis techniques v[illegible] they have certain common features:

a. Data assimilation and entry. This is normally done by engineering aides, keypunch operators, or computer assistants. It will also require some engineering time to organize and supervise the effort. A time estimate can generally be made by estimating the number of data entries involved including any verification time.

b. Computer or manual data processing to produce usable working materials for the analyst, such as, reduced network schematics, network trees, assembly code flow diagrams, and timing diagrams. This is likely to vary so much with different Sneak Analysis techniques that no useful guidance can be given.

c. Detailed analysis by a trained sneak analyst who applies certain "clues" to isolate potential sneak conditions. This is generally done on a worksheet of some sort to aid in the "housekeeping" necessary to assure completeness. It may be done with the assistance of computerized aids. The time required can normally be estimated from the expected number of hours per worksheet. It should be remembered that this is the step in the analysis process most affected by tailoring. Tailoring will result in the analyst reviewing fewer networks and worksheets, thus reducing the amount of analysis time required. It would be expected that tailoring would result in a significant deviation from the linear parts-count relationship presented in Table B 1.

d. Report preparation costs should include technical, typing, editing, and drafting labor, and any special equipment and materials cost required to meet specific CDRL requirements.

2.1.3 Taking advantage of available data. The process of cross-checking a supplier's estimate can become quite involved. The Government monitors should take advantage of all available data sources to make their estimates as accurate as possible. Depending on the situation, the technical or contract monitor may have available the supplier's labor rates, overhead, G&A, and fee structure. This information would be available, for instance, if they were evaluating a supplier's quote on any cost-reimbursable type contract. On fixed fee or incentive fee type contracts, they would also have the supplier's estimate of total man-hours in each labor category, computer, and other direct costs. If the analysis effort were to be funded in phases, they would also have the supplier's estimate by phase. Lacking this specific information on supplier costs, the monitors can use average labor rates in the geographical area involved which are available from the Department of Labor Statistics. Approximate rates for overhead, G&A, and fee structures can be found in other contracts with the involved company or inferred from similar information from competitive companies.

2.1.4 Costs of subcontracting Sneak Analysis. In addition to the costs involved in duplicating the data base at a subcontractor's facility, standard industry practice is for the prime contractor to add G&A and profit charges on a subcontracted Sneak Analysis. Subcontractor costs will already include the subcontractor's G&A and fee charges. This duplication in charges will increase costs and may dictate a direct contract between the Government and the performing activity in some instances, but this consideration must be traded off against other factors, such as which activity is best positioned to manage and understand the technical aspects, and costs involved in incorporating design changes as a result of the analysis.

2.1.5 Procuring activity costs. In addition to contractor costs for Sneak Analysis, the costs for procuring activity coordination must also be included. These costs would include any special costs for travel, coordination, data acquisition, review, or independent technical consultant services associated with the Sneak Analysis effort. The roles of the procuring activity are presented in Section 3.4.4.

APPENDIX C

EXAMPLE

STATEMENT OF WORK

FOR

HARDWARE SNEAK CIRCUIT ANAL

STATEMENT OF WORK
HARDWARE SNEAK CIRCUIT ANALYSIS

1. GENERAL

A Sneak Circuit Analysis shall be performed on the ____________________ hardware. The analysis shall be performed using the sneak analysis network tree approach. This analysis shall identify latent electrical circuit paths and conditions that can cause an unwanted function to occur or inhibit a desired function without component failure. Recommendations for corrective action to eliminate these conditions shall be provided. Also, document errors and areas of design concern discovered during the analysis shall be reported.

2. SCOPE

The ______________________ System/Subsystem of the ________________ shall be analyzed at the detailed component level to identify potentially undesirable circuit conditions. The system(s) included in this analysis are ____ ________________________ and ______________________________ as defined by drawing(s) ____________________________ and ________________.

3. CHANGE ANALYSIS (OPTIONAL)

The analysis shall include identified changes to the data baseline received prior to ___________________, 19__.

The change analysis shall be limited to a total equal to _____ percent of the baseline hardware design data base as established by actual count of data records added, deleted, or revised due to engineering changes. Each change shall be evaluated and subject to separate negotiation depending upon size and complexity.

4. TASK DESCRIPTION

Specific tasks to be performed as part of this analysis contract shall consist of the following:

4.1 Receive and set up files for wiring diagrams, schematics, wire lists and other input data defining electrical continuity, operation and functions of the system(s) to be analyzed.

4.2 Convert circuit data for entry into network trees and apply existing analysis techniques to identify all continuity paths.

4.3 Perform a sneak circuit analysis on the resulting network trees to identify potential sneak circuit conditions, such as:

4.3.1 Sneak paths, wich may allow current or energy to flow along an unexpected route.

4.3.2 Sneak timing, which may cause current or energy to flow or to inhibit a function at an unexpected time.

4.3.3 Sneak indications, which may cause an ambiguous or false display of operating conditions.

4.3.4 Sneak labels, which may cause incorrect stimuli to be initiated.

5. REPORTS

The reports shall be delivered in accordance with the attached schedule. (See Figure 1).

5.1 Prepare Sneak Circuit Reports (SCR) on all potential sneak conditions found. Each report shall describe the sneak condition in detail. The SCA shall include a sketch of the suspect circuit, where appropriate. Recommendations for corrective action shall be given along with reference to supporting documentation. The report forms include a section for the customer to indicate the action taken to resolve the condition being reported. All reports shall be appropriately dated, titled and numbered for indexing and tracking.

5.2 Prepare Design Concern Reports (DCR). The DCR shall describe certain undesirable circuit conditions found during the analysis which do not qualify as sneak circuits. Conditions to be reported include:

- o Single failure points.
- o Unnecessary circuitry or components.
- o Improper implementation of redundancy.
- o Improper application of components.
- o Lack of transient suppression or improper suppression for inductive loads.

5.3 Prepare Drawing Error Reports (DER) on discrepancies found in the input data for the analysis. Each report shall identify the discrepant document and explain the error relative to referenced supporting documentation.

5.4 Prepare and submit activity reports to describe the work accomplished during the reporting period. The reports will include analysis progress, problems, recommendations, and results of meetings. The reports shall be submitted in accordance with the attached schedule (see Figure 1). The SCR, DCR, and DER reports generated during the reporting period shall be attached. A tabulation of all previously submitted reports (SCR, DCR, and DER) including status, shall be attached to each activity report. The status of each report shall be based on the contractor's comments stated on each report.

5.5 Prepare and submit a Final Report containing a summary of the analysis effort, including the general analysis method used, the extent of the analysis, conclusions drawn and recommendations based on the analysis. All SCR, DCR, and DER reports written shall be included. One report shall be prepared at the completion of the baseline analysis, and revised at the end of change analysis, if elected, to include the reports from changes.

6. DATA REQUIREMENTS

The analysis shall be based upon data defining details of electrical continuity and components. The information will be supplied to the Sneak Analysis Contractor in the form of manufacturers detail electrical schematics, wire lists, wiring interconnect diagrams, and component specifications. The data will be supplemented with available functional flow or integrated schematics, interface control documents, and design criteria specifications. Manufacturing level electrical data and supplemental electrical data shall be furnished by ________________________________. Requests for additional data after the baseline master files have been established will be made only for data absolutely essential to the completion of the analysis. Data must be delivered by ____________________, 19__, to enable a timely and accurate analysis.

Delay in receipt of data shall result in a day-for-day slide in the schedule, and the contract price shall be equitably adjusted to reflect additional costs, if any.

7. PERIOD OF PERFORMANCE

The period of performance for the Sneak Circuit Analysis of the ____________________________ System shall be _______ months after receipt of input data necessary to establish the master file beginning ________________, 19___, and ending ________________, 19___. Change analysis shall be performed on all previously described changes received prior to ___________________, 19____, and the final report will be submitted by ___________________, 19____.

APPENDIX D

EXAMPLE
STATEMENT OF WORK
FOR
SOFTWARE SNEAK ANALYSIS

STATEMENT OF WORK
FOR SOFTWARE

1. GENERAL

A Software Sneak Analysis shall be performed on the ______________________ System. The analysis shall be performed using the sneak analysis network tree technique as referenced in Section 3.4.2.2. This analysis shall identify latent data paths and logic conditions that can cause an unwanted function to occur or inhibit a desired function without a hardware failure. Recommendations for corrective action to eliminate these conditions shall be provided. Also, document errors and areas of design concern discovered during the analysis shall be reported.

2. SCOPE

The Software Sneak Analysis shall cover the software program for the ______ ______________________ computer. The software shall be analyzed at the detailed instruction level. The software program is coded in ___________________________.

3. CHANGE ANALYSIS (OPTIONAL)

The analysis shall include changes due to corrective action taken to eliminate Sneak Analysis identified problems received prior to _________________, 19____. The change analysis shall be limited to a total equal to ___% of the baseline design data base as established by actual count of software instructions added, deleted or revised due to corrective changes. Wherever possible all changes shall be submitted by copies of the completed software problem/change reports, copies of the revised data tapes or card decks, and listings.

4. TASK DESCRIPTIONS

Specific tasks to be performed as part of this analysis contract shall consist of the following:

4.1 Receive and set up files for the software program and any design and program documentation defining the operation and functions of the software to be analyzed.

4.2 Process software instructions for entry into computerized algorithms which reduce a software program into topological network trees identifying all data and logic continuity paths.

4.3 Perform a sneak analysis on the resulting software network trees to identify potential sneak conditions, such as:

4.3.1 Sneak paths, which may allow data or logic to flow along an unexpected route.

4.3.2 Sneak timing, which may cause data or logic to flow, or to inhibit a function at an unexpected time.

4.3.3 Sneak indications, which may cause an ambiguous or false display of system operating conditions.

4.3.4 Sneak labels, which may cause incorrect stimuli to be initiated.

5. REPORTS

Reports will be submitted in accordance with the attached schedule. (See Figure 1.)

5.1 Prepare Sneak Software Reports (SSR) on all potential sneak conditions found. Each report shall describe the sneak condition in detail. The SSR shall include a listing of the suspect software instructions where appropriate. Recommendations for appropriate corrective action shall be given along with reference to supporting documentation. The report forms include a section for the customer to indicate the action taken to resolve the condition being reported. All reports shall be appropriately dated, titled and numbered for indexing and tracking.

5.2 Prepare Software Design Concern Reports (SDCR) to describe certain items of concern with specific design implementation. These conditions to be reported include:

5.2.1 Questionable design practices.

5.2.2 Unnecessary software instruction.

5.2.3 Unused software instructions.

5.2.4 Specifications not met or not clear.

5.3 Prepare Software Document Error Reports (SDER) on discrepancies found in the input data for the analysis. Each report shall identify the discrepant document and explain the error relative to referenced supporting documentation.

5.4 Prepare and submit activity reports to describe the work accomplished during the reporting period. The reports will include analysis progress, problems, recommendations, and results of meetings. The reports shall be submitted in accordance with the attached schedule. (See Figure 1.) The SSR, SDCR, and SDER's generated during the reporting period shall be attached. A tabulation of all previously submitted reports (SSR, SDCR, and SDER) including status, shall be attached to each activity report. The status of each report shall be based on the contractor's comments stated on each report.

5.5 Perform analysis of software changes, provided a change analysis is elected, and submit appropriate reports.

5.6 Prepare and submit a Final Report containing a summary of the analysis effort, including the general analysis method used, the extent of the analysis, conclusions drawn and recommendations based on the analysis. All SSR's, SDCR's, and SDER's written shall be included. One report shall be prepared at the completion of the baseline analysis and if a change analysis is performed, the report shall be revised to include any additional reports resulting from the change analysis.

6. DATA REQUIREMENTS

The analysis will be based on the assembly source listing and assembly source code which should be provided on magnetic tape. If the analysis is to be based on a high order language, then the source program listing, high order source code and an assembled program listing should be provided. All reference manuals for the computer, cross-assembler, language and operating system should be made available. Also, it is highly desirable that other program documentation be provided such as module descriptions, flow diagrams and data structure definitions so that the potential system impact for problems found can be more accurately assessed. The above data will be furnished by ____________________. All data must be delivered by ____________________ to enable a timely and accurate analysis. Delay in receipt of data will result in a day-for-day slide in the schedule, and the contract price shall be equitably adjusted to reflect additional costs, if any.

7. PERIOD OF PERFORMANCE

The period of performance for the Sneak Analysis of the ______________ ______________ System shall be _______ months after receipt of input data necessary to establish the configuration as defined in Paragraph 2, SCOPE. Change analysis shall be performed on all previously described changes received prior to ________________, 19__, and the final report shall be submitted by ________________, 19__, provided the contractor has acknowledged all reports by having signed each report and stated the action taken to correct the reported problem.

APPENDIX E

EXAMPLE
STATEMENT OF WORK
FOR
INTEGRATED HARDWARE/SOFTWARE SNEAK ANALYSIS

STATEMENT OF WORK
INTEGRATED HARDWARE/SOFTWARE SNEAK ANALYSIS

1. GENERAL

An Integrated Sneak Analysis shall be performed on the ________________ hardware and software. The analysis shall be performed using the sneak analysis network tree techniques. The Sneak Circuit Analysis shall identify latent electrical circuit paths and conditions that can cause an unwanted function to occur or inhibit a desired function without component failure. The Sneak Software Analysis shall identify latent data paths and logic conditions that can cause an unwanted function to occur or inhibit a desired function without a hardware failure. The integrated analysis shall integrate the interactions of the hardware with the system software. Recommendations for corrective action to eliminate sneak conditions shall be provided. Also, Document Errors and areas of Design Concern discovered during the analysis shall be reported.

2. SCOPE

2.1 Hardware

2.1.1 A Sneak Circuit Analysis shall be performed at the detailed component on the ________________ System of the ________________ to identify potentially undesirable circuit conditions. The subsystems to be analyzed are ________________.

2.1.2 An analysis shall be performed on all interconnections between and within the above subsystem, together with the "interface" functions of the subassemblies/subsystems.

2.1.3 The "interface" function is to be analyzed to its termination inside the subassembly/subsystem. The function will be considered terminated if it connects to (1) chassis or signal ground; (2) a power source; or (3) an electrical element which changes the characteristic or nomenclature of the function.

2.1.4 The Sneak Circuit Analysis shall be performed at the detailed source program level on the ________________ Subsystems as defined by the following drawings: ________________.

2.2 Software

The Sneak Software Analysis shall include the computer programs for the ________________. The ________________ program consists of (number) lines of (Higher Order) Code/Instructions and (number) lines of (Assembly) Code/Instructions.

3. CHANGE ANALYSIS (Optional)

3.1 The hardware change analysis shall include identified changes to the design data baseline received within ________ months after the project start date. The design data baseline is defined in paragraphs 2.1.4 and 2.2.

3.2 The software change analysis shall be limited to ___ percent of the design data baseline. All changes submitted after the ___ month deadline above will be evaluated and subject to separate negotiation, depending upon size and complexity.

4. TASK DESCRIPTION

Specific tasks to be performed as part of this analysis contract shall consist of the following:

4.1 Hardware

4.1.1 Receive and set up files for wiring diagrams, schematics, wire lists and other input data defining electrical continuity, operation and functions of the system(s) to be analyzed.

4.1.2 Convert circuit data for entry into computer-generated, manually drawn network trees. Apply existing analysis techniques to the network trees, to identify all continuity paths.

4.1.3 Perform a sneak circuit analysis on the resulting network trees to identify potential Sneak Circuit Conditions, Design Concerns and Drawing Errors. Design Concerns are defined in paragraph 5.2.1, Drawing Errors in paragraph 5.3. Sneak Circuit Conditions include:

a. Sneak paths, which may allow current or energy to flow along an unexpected route.

b. Sneak timing, which may cause current or energy to flow or to inhibit or initiate a function at an unexpected time.

c. Sneak indications, which may cause an ambiguous or false display of operating conditions.

d. Sneak labels, which may cause incorrect stimuli to be initiated.

4.2 Software

4.2.1 Receive and set up files for the software program, design specifications, logic flow diagrams, and any design and program documentation defining the operation and functions of the software to be analyzed.

4.2.2 Convert software instructions for entry into computerized algorithms which reduce a software program into topological network trees identifying all data and logic continuity paths.

4.2.3 Perform a sneak analysis on the resulting network trees to identify potential Software Sneak Conditions, Sfotware Design Concerns, and Software Document Errors. Software Design Concerns are deifned in paragraph 5.2.2, the Software Document Error in paragraph 5.3, Software Sneak Conditions include:

a. Sneak paths, which may allow data or logic to flow along an unexpected route.

b. Sneak timing, which may cause data or logic to flow, or to inhibit or initiate a function at an unexpected time.

c. Sneak indications, which may cause an ambiguous or false display of system operating conditions.

d. Sneak labels, which may cause incorrect stimuli to be initiated.

4.3 Hardware/Software Integration

In order to provide visibility of interactions of the system's hardware and software, an Integration Analysis shall be performed. The effect of a control operation initiated by a hardware element shall be traced through the hardware until it impacts the system software. Similarly, the logic sequence of a software initiated action shall be followed through the software and hardware until its eventual system impact is assessed.

5. REPORTS

All reports described below shall be prepared and submitted in accordance with the periodic status report requirements shown in the attached project schedule. The reports shall be delivered incrementally with the activity report discussed in paragraph 5.4.

5.1 Prepare Sneak Circuit Reports (SCR) and Sneak Software Reports (SSR) on all potential sneak conditions found. Each report shall describe the sneak condition in detail. The SCR shall include a sketch of the suspect circuit, and the SSR shall include a listing of the suspect software instructions, where appropriate. Recommendations for corrective action shall be given, along with reference to supporting documentation. The report forms shall include a section for the customer to indicate the action taken to resolve the reported condition. All reports shall be appropriately dated, titled and numbered for indexing and tracking.

5.2 Prepare Design Concern Reports (DCR) and Software Design Concern Reports (SDCR).

5.2.1 The DCR shall describe certain undesirable circuit conditions found during the analysis which do not qualify as sneak circuits. Conditions to be reported include:

a. Single failure points.
b. Unnecessary circuitry or components.
c. Improper implementation of redundancy.
d. Improper application of components.
e. Lack of transient suppression or improper suppression for inductive loads.

5.2.2 The SDCR shall describe certain items of concern with specific design implementation. Conditions to be reported include:

a. Questionable design practices.
b. Unnecessary software instructions.
c. Unused software instructions.
d. Specifications not met or not clear.

5.3 Prepare Drawing Error Reports (DER) and Software Document Error Reports (SDER) on discrepancies found in the input data for the analysis. Each report shall identify the discrepant document and explain the error relative to referenced supporting documentation.

5.4 Prepare and submit activity reports to describe the work accomplished during the reporting period. The reports will include analysis progress, problems, recommendations, and results of meetings. The reports shall be submitted in accordance with Figure 1. The SCR, SSR, DCR, SDCR, DER, and SDER reports generated during the reporting period shall be attached. A tabulation of all previously submitted reports (SCR, SSR, DCR, et. al.) including status, shall be attached to each activity report. The status of each report shall be based on the contractor's action taken to resolve each reported condition.

5.5 Prepare and submit a Final Report containing a summary of the analysis effort, including the general analysis method used, the extent of the analysis, conclusions drawn and recommendations based on the analysis. All SCR, SSR, DCR, SDCR, DER, and SDER reports written shall be included. The report shall be prepared and submitted in accordance with Figure 1.

6. DATA REQUIREMENTS

6.1 Hardware

The analysis shall be based upon data defining details of electrical continuity and components. The information will be supplied in the form of manufacturers detail electrical schematics, wire lists, wiring interconnect diagrams, and component specifications. The data will be supplemented with available functional flow or integrated schematics, interface control documents, and design criteria specification. The above information and data shall be furnished prior to the project start date. The required data are listed in Attachment 1. Requests for additional data after the baseline masterfiles have been established will be made only for data absolutely essential to the completion of the analysis. These additional data must be delivered within 30 days after the project start date, to enable a timely and accurate analysis.

6.2 Software

The analysis shall be based upon the software source code as described in paragraph 2.2. All reference manuals for the computers, languages, and operating systems will be made available. Specifications for the software, and for the interface between th_ software and hardware, are required for the integrated analysis. Also, it is highly desirable that other program documentation be provided such as module descriptions, flow diagrams, and data structure definitions so that potential system impact for problems found can be more

accurately assessed. These data will be furnished prior to the project start date. Any additional data must be delivered 30 days after the project start date, to enable a timely and accurate analysis.

6.3 Schedule Impact

Delay in receipt of either hardware or software data shall result in a day-for-day slide in the schedule. The contract price shall be equitably adju ted to reflect additional costs, if any.

7.0 PERIOD OF PERFORMANCE

The period of performance for the Sneak Analysis of the system shall be ________ months after contract start date (see Figure 1).

APPENDIX F

EXAMPLE

DATA ITEM DESCRIPTION

DATA ITEM DESCRIPTION	2. IDENTIFICATION NO(S).	
	AGENCY	NUMBER
1. TITLE ANALYSIS, SNEAK CIRCUIT		DI-R-22594
3. DESCRIPTION/PURPOSE The sneak circuit analysis documents the results of analyses performed to verify the absence or presence of hidden flow paths, unexpected outputs or undesirable functions of equipment or software. The analysis will document the analyses to identify, and corrective action proposed, which can eliminate any latent flow paths that could cause unexpected operations during the life of the hardware or software. It details the methodology used in, and the extent and depth of, the analysis.	4. APPROVAL DATE 6 January 1978 5. OFFICE OF PRIMARY RESPONSIBILITY 6. DDC REQUIRED 8. APPROVAL LIMITATION	
7. APPLICATION/INTERRELATIONSHIP 7.1 The sneak circuit analysis provides documentation from which the Government procuring activity can make determinations concerning system and equipment unwanted functions or inhibition of desired functions in the absence of component failure. 7.2 Sneak circuit analysis is applicable to mission critical hardware and software systems and equipment.	9. REFERENCES (Mandatory as cited in block 10) MIL-STD-785B Notice 1 (EC) MCSL NUMBER(S) 23009	

10. PREPARATION INSTRUCTIONS

10.1 The Sneak Circuit Analysis shall include, but is not limited to, the following data:

a. contract number, exhibit line item number and this DID number

b. equipment specification number and paragraph number if applicable

d. description of the methodology and procedures used to satisfy the requirements for sneak circuit analysis as stipulated in MIL-STD-785A Notice 1 (EC)

e. recommendations for corrective action with sufficient detail to demonstrate that the sneak path will be eliminated.

10.2 Analyses shall be in the contractor's own format.

FIGURE 10. Contract data requirements list (Continued)

DD FORM 1 JUN 68 1664 S/N-0102-019-4000 PAGE ___ OF ___ PAGES

APPENDIX G

EXAMPLE

THIRD PARTY DATA AGREEMENT

PROPRIETARY INFORMATION
NONDISCLOSURE AGREEMENT SAMPLE
(THIRD PARTY AGREEMENT)

Proprietary agreements. ______(Company Name)______ (Division) ______(Address)______, and (Company Name) (Division) ______(Address)______ have initiated a coordinated exchange of data relating to ______(Name of Program)______ program. To maximize the effectiveness of the program it appears that information exchange is or may become necessary between the two companies. Accordingly, it is proposed that the following letter agreement be entered into between these two companies to cover all transmittals of information in connection with the program (or any subsequent programs or contracts resulting from the program) between (First Company Name) and ______(Second Company Name)______, and by the United States Government to ______(First Company Name)______ or ______(Second Company Name)______.

1. Each party, to the extent of its right to do so, shall submit to the other party technical information at times and of kinds and in forms which in the judgement of the party originating the information are appropriate to fulfillment of the obligations assumed by that party under its respective portion of the ______(Program Name)______ program. This agreement shall not be construed as itself creating any obligation on either party to furnish information to the other.

2. Any technical information of ______(First Company Name)______ which is submitted to ______(Second Company Name)______ and any technical information of ______(Second Company Name)______ which is submitted to ______(First Company Name)______ under this agreement, either directly or through the United States Government, which information is designated as proprietary to the submitting party by an appropriate stamp, legend or other notice in writing shall be subject to the provisions as to disclosure and use hereinafter set forth. Information initially disclosed orally shall not be deemed proprietary unless such information is confirmed as such in writing by the submitting party within thirty (30) days after the initial disclosure thereof to the recipient party. During the 30-day period, such orally disclosed information shall be protected as if it were proprietary information. Any such information which is not accepted by the recipient party shall be promptly returned to the submitting party. All such information which is accepted by the recipient party shall, for a period of ______ years from the date of transmittal of each item of information covered by this provision:

   a. Be used, duplicated and disclosed by the recipient party solely for the purposes of performance of its portion of these joint activities.

   b. Not be used, duplicated or disclosed for purposes of manufacture or procurement of the equipment to which the information pertains.

c. Be disclosed only to personnel of the recipient party and of the United States Government with appropriate restrictive legends authorized by ASPR: and

d. If reproduced in whole or in part, shall carry a proprietary notice similar to that with which sumbitted to the recipient party.

3. Information shall not be deemed proprietary or confidential, and the recipient party shall have no obligations as to any information which:

   a. Is already known to the recipient party.

   b. Is or becomes publicly known through no wrongful act of the recipient party.

   c. Is rightfully received from a third party without similar restrictions and without breach of this agreement.

   d. Is furnished to the United States Government or other third party by the submitting party without similar restrictions of use and disclosure.

   e. Is approved for release or use by written authorization of the submitting party.

   Provided, however, that information not deemed proprietary by the recipient party in accordance with the above, but nevertheless marked proprietary by the submitting party, will not be disclosed by the recipient party without markings revealing the name or interest of the submitting party.

4. The recipient party shall not be liable for inadvertent, accidental, mistaken disclosure or use by its employees, of information obtained under this government, provided that:

   a. The recipient party shall use the same degree of care as used to protect its own proprietary information of like importance.

   b. Upon discovery of such disclosure or use, the recipient party shall endeavor to prevent further disclosure or use.

5. With respect to any exchange of proprietary or confidential information which may occur as a result of this Agreement, it is expressly understood and agreed that the below listed employees shall, on behalf of the respective parties, be the sole and exclusive individuals authorized to receive and/or transmit proprietary or confidential information under this Agreement:

(First Company Name)	(Second Company Name)

6. As regards the individuals identified in paragraph 5 above, each party shall have the right and power to redesignate such persons within their organizations as are authorized to receive or transmit proprietary or confidential information exchanged under this Agreement. Any such redesignations which are made by either party shall be effected by tendering written notice of such change to the other party.

7. Nothing contained in this agreement shall be construed as granting or conferring any rights by license or otherwise, expressly or implied, for any invention or discovery, or any patent covering such invention or discovery, which is made or acquired prior to or after the date of this Agreement.

8. Any information not designated as proprietary in accordance with Paragraph 2 shall not, unless otherwise specifically agreed upon in writing by the recipient party, be deemed to be proprietary or submitted in confidence and shall be acquired by the recipient party free from any restrictions of use or disclosure (other than a claim for patent infringement).

9. This Agreement, and all rights and obligations established hereby except those specified in Paragraph 10, may be terminated by either party on sixty (60) days written notice to the other. Unless thus earlier terminated, termination will occur upon the first of the following events:

   a. Completion or termination of these joint activities by either party,

   b. The expiration of ________ years from the effective date of this Agreement.

10. Termination of this Agreement shall not relieve the recipient party of the obligations imposed by Paragraph 2 hereof with respect to proprietary information exchanged prior to the effective date of the termination; those obligations shall continue for the period applicable to each item of information as specified in said paragraph.

ACCEPTED:

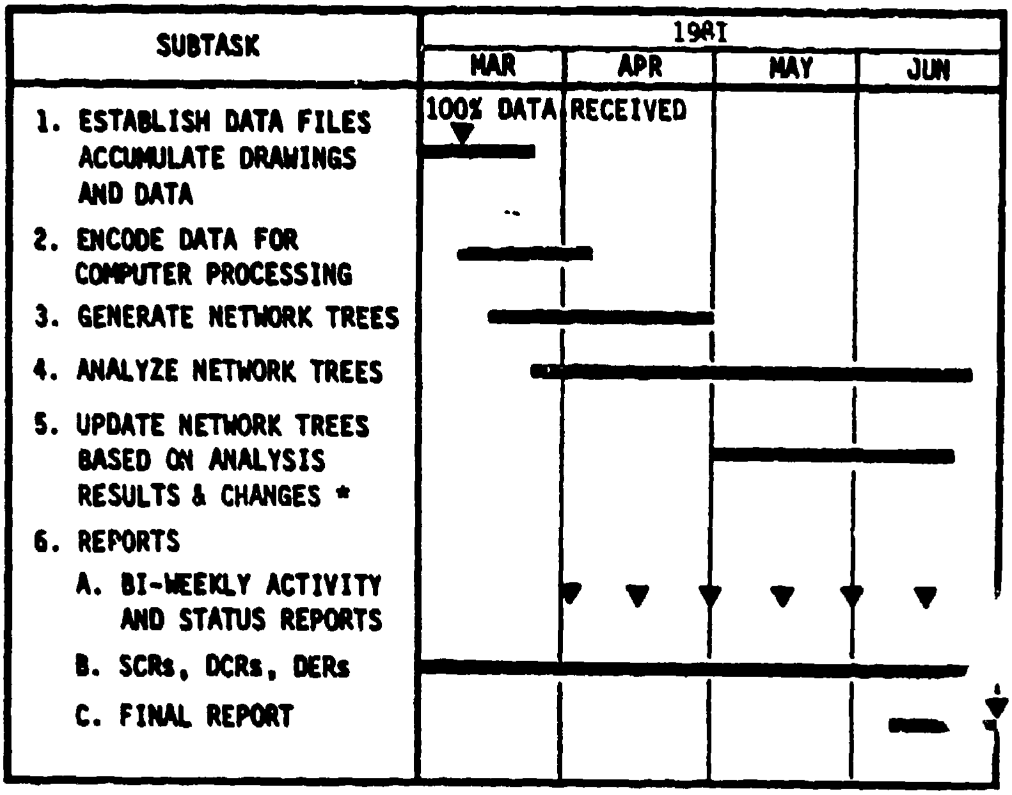
(First Company Name and Address)	(Second Company Name and Address)
Individual: ____________________	Individual: ____________________
Title: ____________________	Title: ____________________
Date: ____________________	Date: ____________________

APPENDIX H

EXAMPLE

PROJECT SCHEDULE

EXAMPLE SNEAK ANALYSIS PROJECT SCHEDULE



* IF CHANGE OPTION SELECTED

APPENDIX I

PERFORMANCE/DESIGN/INTERFACE

SPECIFICATION

FOR

SNEAK ANALYSIS

PERFORMANCE/DESIGN/INTERFACE
SPECIFICATION
FOR
SNEAK ANALYSIS

1. SCOPE

1.1 Scope - This specification covers the design requirements for application of Sneak Analysis (SA) to electrical/electronic and software systems.

1.2 Purpose - The Sneak Analysis (SA) technique described herein establishes a standard procedure for performing the analysis and reporting the results. The analysis identifies and reports all sneak paths, sneak timing, sneak labels, and sneak indicators which may exist in the design. All areas of design concern and document errors discovered during the sneak analysis are also reported. Such sneak conditions and design concerns that are discovered are assessed for their impact on system performance.

1.3 Intended Use - This specification is intended for use as a design requirement for inclusion in contract end item (CEI) specifications, system specifications, document procedures, and/or contracts as applicable.

2. APPLICABLE DOCUMENTS

2.1 Documentation - The following documents of the issue noted or, if not noted, the issue in effect as of the date of the contract as shown in Department of Defense Index of Specifications and Standards, and Army Missile Command Index of Purchase Descriptions and Standards, form a part of this specification to the extent specified herein.

MILITARY SPECIFICATIONS

MIRADCOM EXHIBIT QR-800-J (15 June 1978)	Reliability Program for Systems and Equipment Development (Para. 5.2.5)
MICOM-PAM-385-4 (23 March 1973)	Safety, Ignition Systems for Army Rockets and Missiles (Para. 8.b, 8.c)
MIL-STD-822A (28 June 1977)	System Safety Program Requirements (Para. 5.5.1.2.c)
MIL-STD-781C (21 October 1977)	Reliability Design Qualification and Acceptance Test Standard (Appendix A, Para. 40.7)
MIL-STD-785B (21 Aug 1978)	Reliability Program Plan for Systems and Equipment (Para. 5.2.1.2)

3. REQUIREMENTS

3.1 Definitions

3.1.1 Sneak circuit. A sneak circuit is a latent path which causes an unwanted function to occur, or inhibits a desired function, without regard to component failure.

3.1.2 Software sneak. A software sneak is a combination of conditions, causing unplanned events, that exhibit unapparent cause/effect relationships, which may escape detection during system testing and occur without regard to hardware failure.

3.1.3 Kinds of sneaks. The four kinds of sneaks are:

a. Sneak path - initiates an undesired function or prevents a desired function.

b. Sneak timing - is an energy, data, or logic flow which causes or inhibits desired functions at an unexpected time.

c. Sneak indication - is an ambiguous or false display of a condition or data which could result in an undesired action being taken.

d. Sneak label - is an ambiguous or false name or function title which could result in the application of the wrong stimuli by an operator.

3.1.4 Sneak Analysis (SA). SA is a type of engineering analysis performed on an electrical or electronic hardware system, or computer software program. SA is a unique technique which involves accumulation of detailed circuit diagrams, wirelists, and software; arrangement of circuit/software elements into topological nodal sets (network trees) and the examination of these nodal sets for sneak circuits.

3.1.5 Assumptions. Assumptions are made when performing SA to establish the analysis boundaries, define terminology, and keep the scope within cost effective bounds. Tables I and II list the more common assumptions for hardware and software respectively.

TABLE I. HARDWARE SA ASSUMPTIONS

a.	A Sneak Circuit is not dependent on a component or circuit failure.
b.	Unless otherwise specified, signals which cross analysis boundaries (out of scope) are assumed to be correct in voltage, polarity, and time for the circuit being analyzed.
c.	The data base for the analysis represents the "as built" configuration of the system.
d.	Parametric calculations are performed only to the extent necessary to understand true circuit operations.
e.	Environmental affects are not normally considered in the analysis.

TABLE II. SOFTWARE SA ASSUMPTIONS

a. The software specification is the design intent of the software.
b. The assembler or compiler does not introduce errors into the software.
c. Assembled or compiled software is free of syntax errors, ie., typographical errors.
d. The data provided represents the complete software program under consideration.
e. Hardware induced problems are not considered.

3.1.6 Design concern. A design concern is a hardware or software condition which is identified during the Sneak Analysis process which could cause or result in a failure, a marginal operation, or a hazardous situation. The following are kinds of design concerns:

a. Unnecessary logic, components, or circuits.
b. Improper sequence of software instructions.
c. Single failure points.
d. Unnecessary power consumption.
e. Improper or marginal application of components.

3.1.7 Drawing or document error report (DER). A DER is one prepared during the Sneak Analysis process which notifies the procuring activity of a discrepancy found in the furnished data. The discrepancy can be within a single document or between two or more documents.

3.2 Scheduling SA in a System's Life Cycle - The typical life cycle for production programs or systems is shown in Figure I-1. A detailed component level SA shall be performed when good engineering documentation and drawings are released to manufacturing. The time period prior to and just after the Critical Design Review (CDR) milestone is typically the best time in the life cycle to start the analysis. Performing SA from CDR to any later development phase including the unlimited production and deployment phases is usually justified because problems occur which were not evident during full scale development. Each program or system shall be evaluated for its documentation maturity, and schedule forcing functions, to determine when to start SA in the most cost-effective manner.

3.2.1 Scheduling of Software Sneak Analysis. The performance of a detailed software Sneak Analysis also requires mature data. Although software Sneak Analysis does not require the execution of the code, the code shall be debugged, as a minimum, and the normal development validation and verification process initiated when starting the software SA. Like hardware SA, the time period CDR is typically the best time to start a detailed software sneak analysis. Usually several versions of the software program are developed to agree with the needs of hardware factory testing, field testing, and operational use. The software Sneak Analysis techniques used shall be adaptable to handle these program changes.

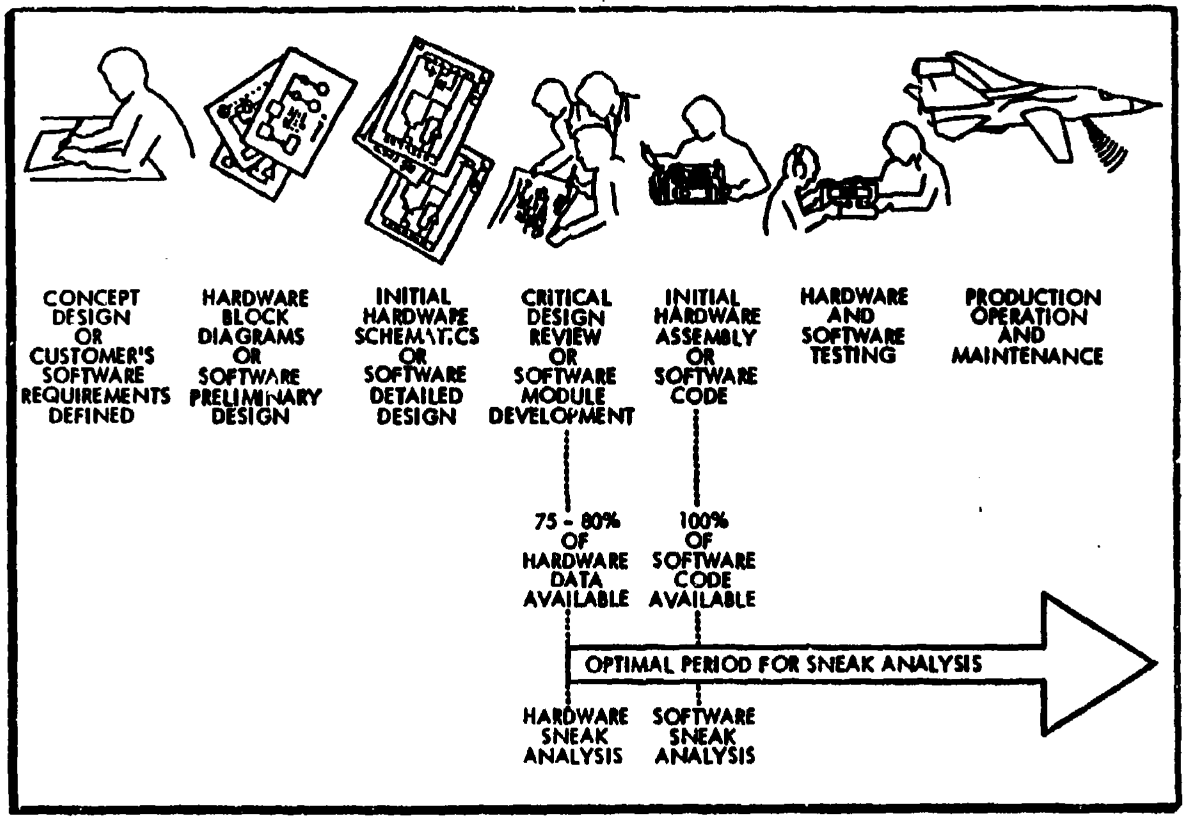


Figure I-1. Program Life Cycle

3.3 Sneak Analysis Performance. The following elements and procedures are required to perform a valid and cost-effective sneak analysis.

a. Collecting the data base.
b. Partitioning of the system.
c. Organizing and structuring of the data base.
d. Application of SA techniques.
e. Reporting of the analysis.
f. Quality control (QC) review of data and the analysis.
g. Control of data management and change.
h. Technical interface with the procuring activity's representative.
i. Delivery and debriefing of final report.

3.3.1 Collecting the data base. The performance of SA requires the organization of the data and certain other sequential steps required to process the data for analysis. The data furnished for the analysis shall be reviewed for completeness and an overview block diagram prepared. The data shall be further organized, logged, and filed for use. Data requests shall be processed for any missing data or information deemed necessary to support the analysis.

3.3.2 Partitioning of the system. The hardware or software system shall be partitioned, subdividing it into modules, functional circuits, subroutines, or manageable size pieces. A properly partitioned system will result in functional modules in a topological network tree format. All network trees shall be functionally oriented, providing an easy application of clue lists. Partitioning also provides an easily accountable and flexible system to analyze. Grounding trees or power distribution trees shall be analyzed separately. (Typical areas of partitioning are busses, both power and ground; cross ties between redundant circuitry; fuses; circuit breakers; and software functional modules.)

3.3.3 Organizing the data. The data selected for the analysis shall be organized to provide a complete search of all paths in the software or hardware system. All information about this system shall be completely and accurately structured. Structuring is the input which establishes the data base masterfile. This masterfile represents the complete data necessary to describe the system to be analyzed. The structuring technique permits the analysts to accurately and uniquely link the components and circuit segments together to form a complete electrical path. In the case of software, it shall be a complete logic and data path. All paths shall be uniquely identified. These paths shall be used to draw the network trees. The network trees shall contain all of the necessary information to apply the SA technique, and represent the system accurately.

3.3.4 Application of SA technique. The network trees shall be drawn so as to form one or more of the five basic topological patterns shown in Figure I2. Recognition of these patterns is an important step in the analysis. Using these patterns, analyst shall apply sneak condition criteria, or "clues," to the circuitry. When all the sneak condition criteria applicable to a particular pattern have been considered, a high degree of confidence is established that all possible sneak conditions resulting from that portion of the circuitry have been identified. SA training shall be provided the analyst, to allow the analyst to quickly evaluate all modes of circuit operation, including the functional interfaces with other circuitry. A detailed analysis of each identified condition

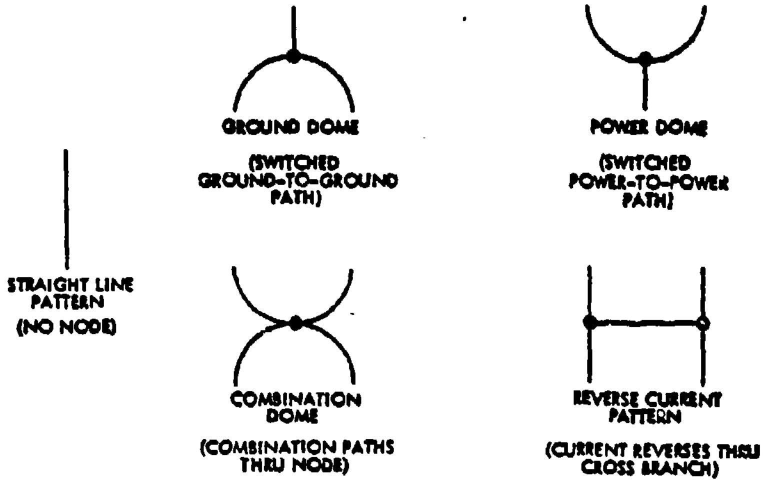


Figure I-2 Topological Patterns

shall be performed to determine the cause and effect of that condition. This evaluation shall result in the reporting of sneak circuits, design concerns, and drawing errors to the system designer for verification and resolution. Design Concern and Drawing Error Reports are incidental outputs of the application of the SA technique.

3.3.5 Reporting of the analysis. Periodic reports shall be generated. The report shall be written using standard formats and delivered to the contracting agency. Formats for Sneak Circuit Reports (SCR's), Sneak Software Reports (SSR's), Design Concern Reports (DCR's), Software Design Concern Reports (SDCR's), Drawing Error Reports (DER's), and Software Document Error Reports (SDER's) are shown in Figures 3 through 8. The "potential impact" and "recommendation" parts of the reports reflect the SA analyst's assessment of the findings. The "explanation" part of the report contains references to attached figures and tables and describes the total intent of the report and analysis findings. A status sheet reports the verification and resolution activity performed by the contracting agency in response to the sneak analysis reports until they are closed out. A final report shall be prepared at the conclusions of the analysis and shall contain a description of the analysis activity, copies of all reports generated during the analysis, a current status sheet, and other pertinent information related to the analysis and its findings.

3.3.6 Quality control (QC). Throughout the analysis period a QC engineer shall review the network trees, proposed reports of conditions found, and the master data file. The QC engineer shall be thoroughly trained and highly skilled in the techniques and methods of performing an SA. Near the end of the analysis, and before writing the final report, senior SA engineers shall review all nodal sets, network trees, completed reports, and related data to establish that a complete and thorough analysis has been accomplished.

3.3.7 Data management and change control. The initial data establishes the baseline data file (masterfile). The masterfile represents the data base for the complete analysis. Data changes received after this time shall be added into the masterfile and incorporated into the applicable network trees. Those trees shall be reanalyzed to determine that the change accomplished its intent and that no new sneaks were introduced.

3.3.7.1 Configuration management. The data base established for SA represents the system "as-built" or "as-coded" source. This data base shall be used for configuration management of the design. Changes made to the design or block modifications shall be incorporated into the data base. This provides a continuing management tool to control each configuration.

3.3.7.2 SA change control. Projects having block configuration changes shall require SA to be performed on each configuration. This is possible because the SA masterfile and all related documentation shall be stored for a minimum of 3 years, per ASPR 7-104.15. A contractual request for SA on a block update shall be implemented by incorporating the drawing changes or new documentation into the stored masterfile. Old network trees shall be modified or replaced as necessary for analysis.

3.3.8 Interfaces with the Technical Monitor. A government contract for performing SA usually specifies a contracting officer and a technical monitor. The technical monitor provides a day-to-day contact for information and data transmittal as necessary. The technical monitor shall receive the SA reports for investigation, resolution, and feedback.

3.3.9 Final Report Delivery and Debriefing. The final report as described in Section 3.3.5 shall be prepared near the end of the analysis period after the QC effort is complete and satisfactory. Sufficient time after delivery of the report shall be allowed for the contracting agency to review it and prepare comments for the debriefing meeting. The debriefing meeting shall resolve the outstanding open reports or establish a report date for closeout.

4. QUALITY ASSURANCE PROVISIONS

4.1 Accountability.

4.1.1 General. To assure a valid Sneak Analysis, provisions must be established to insure that: (1) all paths within a system have been analyzed; (2) each path is directly traceable to the network tree in which it was analyzed; (3) each component/statement is directly traceable to the path in which it was analyzed; and (4) each component/statement is directly traceable to the specific documentation used to establish the data base masterfile.

4.1.2 The following provisions shall be used to assure a valid SA:

4.1.2.1 Network trees analyzed for sneak circuits shall be traceable to the system's manufacturing drawings or source code. The network trees shall contain all the wiring, components, or statements used to generate the tree. Further, all paths necessary to initiate and complete a given function shall be shown or referenced on one network tree.

4.1.2.2 Each network tree shall be independently numbered.

4.1.2.3 An index shall be developed to show in which network tree each component or statement appears.

4.1.2.4 Each path shall be traceable to the network tree in which it appears.

4.1.2.5 Each path shall be independently numbered, and its wire segments, components, or statements traceable to the system's manufacturing drawings or source code in which they appear.

4.1.2.6 Each path shall be independently analyzed as to its effect on system operation, and records maintained indicating analysis results and analyst.

APPENDIX J

EXAMPLE SNEAK ANALYSIS REPORTS

SNEAK CIRCUIT REPORT -1

TITLE SNEAK CURRENT PATH RESULTS IN UNINTENTIONAL MASTER ARMING OF WPN RELEASE SQUIB FIRING CIRCUITS

REFERENCES

1. Groundrush AVN Dwg. No. 501741, Rev. A, "Schematic Diagram-Armament Panel"
2. Groundrush AVN Dwg. No. 501608, Rev. C, "Interconnect List, Wpn Cntrlr"
3. Groundrush AVN Dwg. No. 501476, Rev. A, "Circuit Card Assy - Release-A2,A3"
4. Groundrush AVN Dwg. No. 501233, Rev. A, "Circuit Card Assy-Wpn Interface"
5. Groundrush AVN Dwg. No. DW50100-302, Rev. A, "Wiring Harness, W302"
6. Groundrush AVN Dwg. No. DW50199-401, Rev. B, "Wiring Harness, W401"
7. Coil Winders Inc. Dwg. No. 56R-8-41, Rev.-, "Box, Relay WPN-9421A1"

MODULE/EQUIPMENT

WEAPON CONTROLLER (9431A2)

EXPLANATION

As shown in Figure 1, when the Master Arm switch is off, Emergency Jettison has not been selected, and the Weapon Select switch is left in the Center Station position, a sneak path exists from the +28VDC Weapon Control power through the Weapon Select Switch (9417A3S3) through 9431A2A1R1 to charge capacitor 9431A2A1C1 and then through transistor 9431A2A1Q1 to the firing circuit. This bypasses the Master Arm 'A' function. Similar paths exist for Master Arm 'B' and the Left and Right Wing Stations.

POTENTIAL IMPACT

1. Unexpected Master Arm power may contribute to inadvertent weapon release.

2. The function of the Weapon Release 'A' and 'B' circuit breakers (2456A1CB1 and 2456A1CB2) may be bypassed.

RECOMMENDATION

Add a blocking diode as shown in Figure 2.

REPORTED BY J. L. Vogas J. L. Vogas DATE October 16, 1980

CUSTOMER ACTION

SNEAK CIRCUIT REPORT - 1

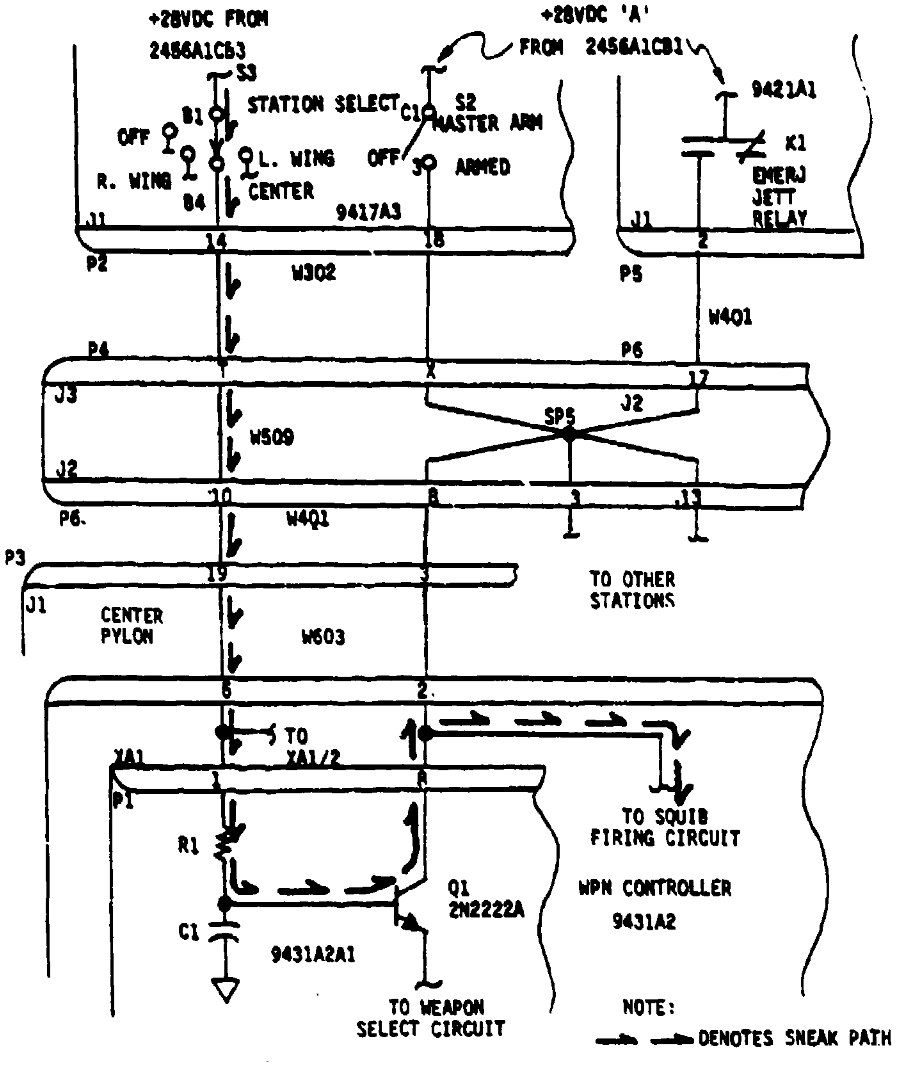


FIGURE 1 SNEAK PATH TO SQUIB FIRING CIRCUIT

SNEAK CIRCUIT REPORT - 1

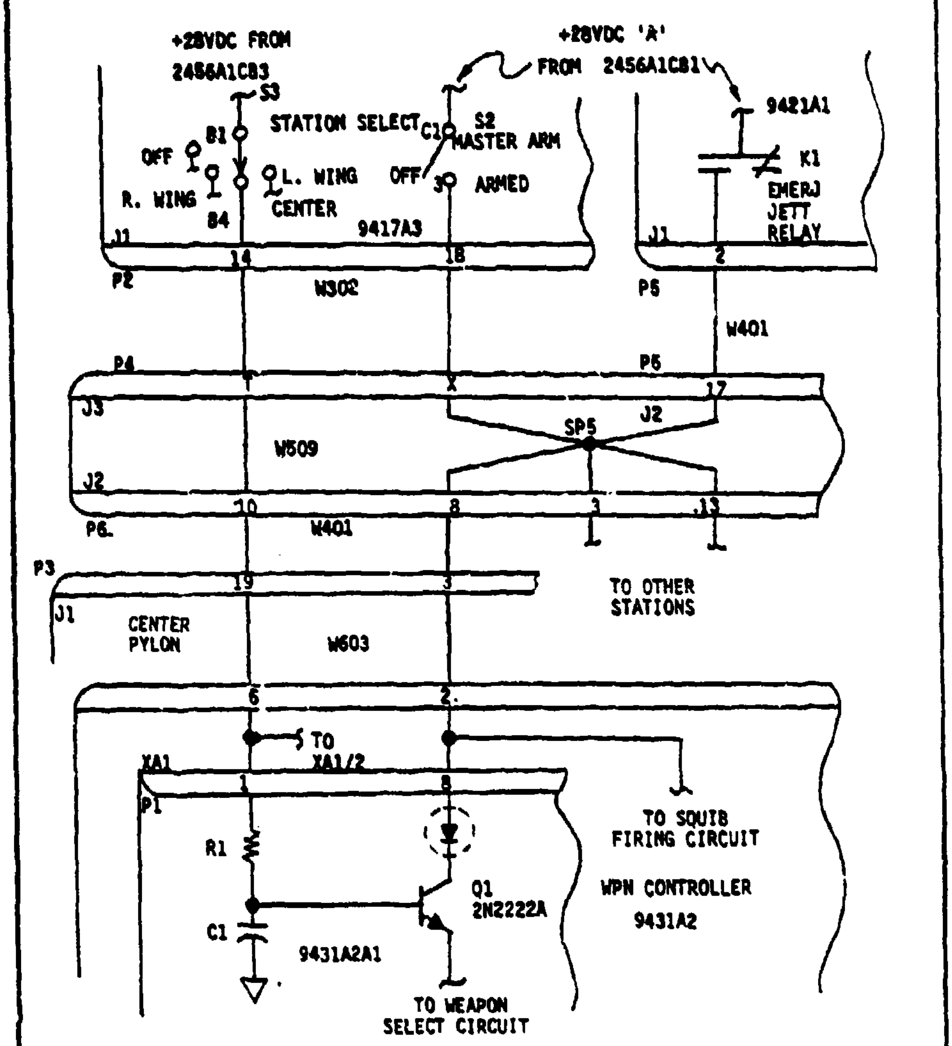


FIGURE 2. RECOMMENDED CIRCUIT MODIFICATION

PROJECT F-99 WCM

DESIGN CONCERN REPORT - 2

TITLE SHARED +5V & +28V RETURN PATHS MAY RESULT IN ERRONEOUS WEAPON DISABLED SIGNAL

REFERENCES

1. Groundrush Avn Dwg. No. DW80199-401, Rev. B, "Wiring Harness, W401."
2. Groundrush Avn Dwg. No. DW80199-603, Rev. A, "Wiring Harness, W603."
3. Groundrush Avn Dwg. No. S01733, Rev. D, "Interconnect List, Stores Cntlr."

MODULE/EQUIPMENT

WIRING HARNESS (W401)

EXPLANATION

As shown in Figure 1, the +28 volt return for the release enable relays (K1 and K2) and the weapon release squibs in the pylon share the same return path through cable W401 with the +5 volt return for the Weapon Disabled buffer (U2) in the weapon. The high +28VDC return current may raise the potential at the ground pin (pin 7) of U2 to cause its output to be falsely seen as a logic 'high' at the input of U9.

POTENTIAL IMPACT

A false indication of WPN DISABLED may be seen by the Stores Controller when Weapon Release Enable or Weapon Release is commanded.

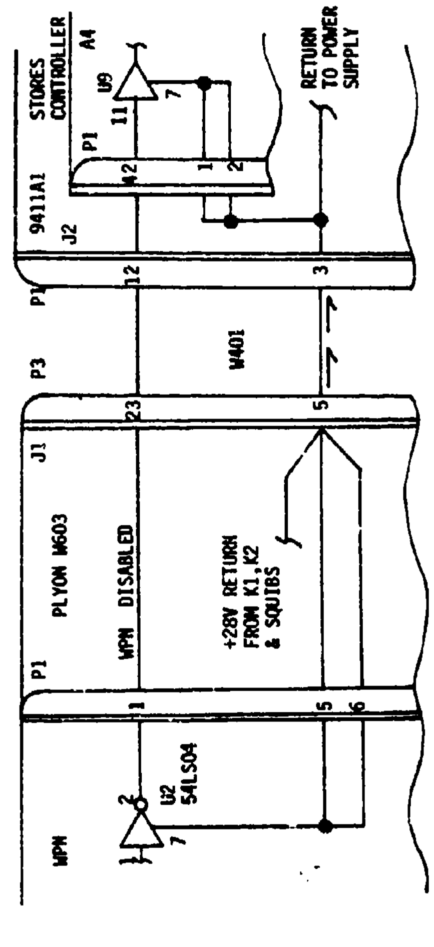
RECOMMENDATION

Provide a separate return path for the +28 volt and +5 volt returns through cable W401 and the Stores Controller.

REPORTED BY J. L. Vogas J. L. Vogas **DATE** 2/23/81

CUSTOMER ACTION

DESIGN CONCERN REPORT - 2



NOTE: DENOTE SHARED +28V AND +5V RETURN

FIGURE 1 WPN DISABLED SIGNAL AND RETURN

PROJECT F-99 WCM

DRAWING ERROR REPORT - 2

DOCUMENT NUMBER	REV	VENDOR	MODULE/EQUIPMENT
501233	A	GROUNDRUSH AVN	PYLON

DOCUMENT TITLE

CIRCUIT CARD ASSY - WPN INTERFACE

REFERENCES

MIL-R-83401/2 (USAF), 2 February 1973.

DISCREPANCY

On the subject document (schematic), device U2 is listed as M8340102-H102FB and shown as in Figure 1A. However, Reference 1 shows the internal connections for this device to be as shown in Figure 1B.

ASSUMED CORRECTION

On the subject document (501233), change pin 14 to pin 16.

REPORTED BY C. A. Hailey *C. A. Hailey* DATE October 16, 1980

CUSTOMER ACTION

DRAWING ERROR REPORT - 2

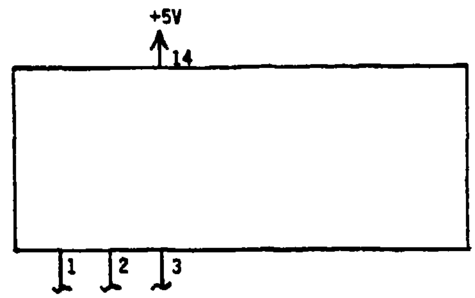


FIGURE 1A - EXTERNAL CONNECTIONS AS SHOWN ON SCHEMATIC (501233)

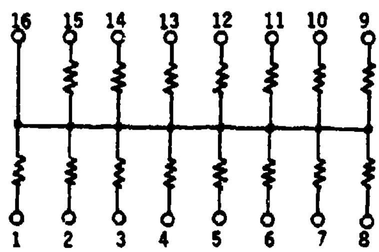


FIGURE 1B - INTERNAL CONNECTION AS SHOWN ON MIL-R-83401/2

PROJECT F-99 WCM PAGE 1 OF 2

SNEAK SOFTWARE REPORT -1

TITLE INCORRECT TEST TO RETARGET WEAPON

REFERENCES

1. Groundrush Document No. 40198, Rev. C, Program Specifications
2. Groundrush Program Source Code, Release 19.1, F-99
3. Groundrush AVN Dwg. No. 501741, Rev. A, "Schematic Diagram - Armament Panel"

MODULE/EQUIPMENT

TXBCP MODULE - TEST-BIT COMPARE

EXPLANATION

When the operator selects the weapon retargeting function from console switches, a value is passed from the RETARGET module to the TEST-BIT COMPARE module. Per Reference 1, the value is stored in variable UART1 and tested in the TEST-BIT COMPARE module. If the value of UART1 = 0, 1, 2, or 3, then program processing continues through the TRUE branch and the selected weapon is retargeted. If the value of UART1 is >3, the software is expected to transfer control to the FALSE branch, display "RETARGET ERROR" on the operator's console, and interrupt processing. Restart is contingent on operator response.

The Reference 2 program source code configuration shown in Figure 1 will not generate the "RETARGET ERROR" message and program interrupt for all values of UART1>3. For example, in an error condition where UART1=8, the test will inhibit the error display and subsequent program interrupt and then incorrectly allow program processing to continue through the TRUE condition into the retargeting logic.

POTENTIAL IMPACT

1. Mission failure due to incorrect targeting of weapons
2. "RETARGET ERROR" display and program interrupt are inhibited for particular conditions.

RECOMMENDATION

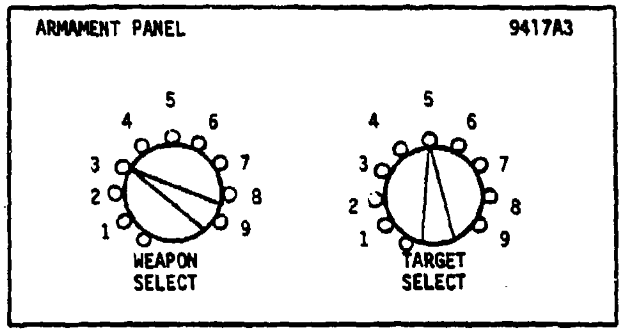
Modify the Reference 2 computer program to accomplish the intent of the Reference 1 specifications. Change the branch instruction to check for a numeric value $0 \leq UART1 \leq 3$. If UART1 is in this range, branch to the TRUE condition and retarget weapon.

REPORTED BY ______________ **DATE** ______________

CUSTOMER ACTION

Program source code has been changed per DCN 1010 to Reference 2. Correction of error reduced incidence of equipment maintenance and RETEST-OKAY dispositions.

SNEAK SOFTWARE REPORT -1



(FROM RETARGET MODULE)

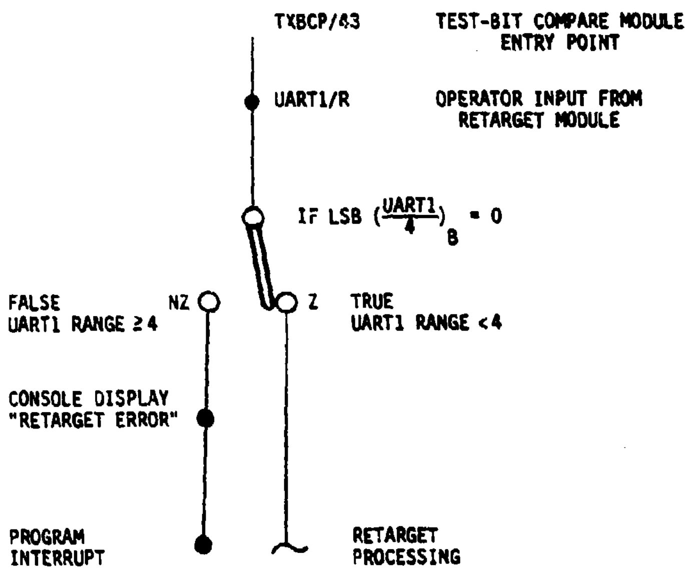


FIGURE 1 - INCORRECT TEST TO RETARGET WEAPON

PROJECT F-99 WCM **PAGE** 1 **OF** 2

SOFTWARE DESIGN CONCERN REPORT -1

TITLE INCORRECTLY ESTIMATED UTILITY ROUTINE PROCESSING TIMES

REFERENCES

1. Groundrush Program Source Code, Release 19.1, F-99
2. Groundrush Document No. 42577, Rev. -, Assembler Manual

MODULE/EQUIPMENT

Refer to Table 1

EXPLANATION

The time to process the Reference 1 software program utility subroutines does not agree with the actual execution times as calculated by a worst-case path summation of the individual instruction execution times listed in the Reference 2 Assembler Manual. These discrepancies are described in Table 1.

POTENTIAL IMPACT

1. Possible timing problems
2. Possible unauthorized or undocumented software code changes

RECOMMENDATION

1. Compare software utility subroutine code to the Reference 2 document and identify differences.
2. Update code and documentation as necessary.

REPORTED BY D. Burath **DATE** 10/10/80

CUSTOMER ACTION

SOFTWARE DESIGN CONCERN REPORT -1

SUBROUTINE	STATED TIME LOADING IN MICROSECONDS	ACTUAL TIME LOADING IN MICROSECONDS	STATEMENT NO. LOCATION
1. START A/D CONVERSION SUBROUTINE	12.93	7.39	5010
2. A/D SIGNAL STORAGE SUBROUTINE	20.20	15.51	5240
3. D/A SIGNAL FORM SUBROUTINE	5.56	7.23	5415
GENERAL 1ST & 2ND ORDER FILTER SUBROUTINE			
4. ● 1ST ORDER	60.00	76.60	5675
5. ● 2ND ORDER	110.00	120.60	5680
6. ● ADD FOR OUTPUT GAIN	20.00	15.30	5685
PURE INTEGRATION SUBROUTINE			
7. ● FORM 1	66.12	64.62	6275
8. ● FORM 2	60.78	57.95	6280
9. ● ADD FOR ASYMMETRICAL LIMITS	18.59	12.84	6285
10. SPECIAL FIRST ORDER SUBROUTINE	122.10	72.05	6565
11. LIMIT, GAIN, SHIFT, SUBROUTINE	18.90	13.90	6795
12. ASYMMETRICAL LIMIT SUBROUTINE	16.09	17.20	6945
13. FILTER START SUBROUTINE	12.71	14.21	7070

TABLE 1. STATED AND ACTUAL PROCESSING TIMES FOR UTILITY SUBROUTINES

PROJECT F-99 WCM PAGE 1 OF 1

SOFTWARE DOCUMENT ERROR REPORT -1

DOCUMENT NUMBER	REV	VENDOR	MODULE/EQUIPMENT
43290	A	GROUNDRUSH	SUBROUTINE CUT

DOCUMENT TITLE F-99 WEAPON CONTROL SOFTWARE DESIGN

REFERENCES

1. Groundrush Program Source Code, Release 19.1, F-99

DISCREPANCY

The computer program design document, page 32, contains a branch in subroutine CUT to BITS. However, page 33 of the software design document provides a flow chart of subroutine CUT with a branch to BYTES. The software listing, Reference 1, lists a branch to BITS at instruction 1100.

ASSUMED CORRECTION

Change BYTES in the software design document flow chart, page 33, to BITS.

REPORTED BY D. Buratti DATE 10/2/80

CUSTOMER ACTION

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.